Journal of Computational Physics **148,** 81–124 (1999)
Article ID jcph.1998.6106, available online at http://www.idealibrary.com on IDEAL®

An Adaptive Level Set Approach for Incompressible Two-Phase Flows[1]

Mark Sussman,*[,2] Ann S. Almgren,† John B. Bell,† Phillip Colella,† Louis H. Howell,† and Michael L. Welcome†

Department of Mathematics, University of California Davis, Davis, California 95616;
†Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, Berkeley, California 94720
E-mail: sussman@math.ucdavis.edu

Received February 6, 1997; revised September 9, 1998

We present a numerical method using the level set approach for solving incompressible two-phase flow with surface tension. In the level set approach, the free surface is represented as the zero level set of a smooth function; this has the effect of replacing the advection of density, which has steep gradients at the free surface, with the advection of the level set function, which is smooth. In addition, the free surface can merge or break up with no special treatment. We maintain the level set function as the signed distance from the free surface in order to accurately compute flows with high density ratios and stiff surface tension effects. In this work, we couple the level set scheme to an adaptive projection method for the incompressible Navier–Stokes equations, in order to achieve higher resolution of the free surface with a minimum of additional expense. We present two-dimensional axisymmetric and fully three-dimensional results of air bubble and water drop computations. © 1999 Academic Press

Key Words: incompressible; two-phase flow; level sets; adaptive mesh refinement.

1. INTRODUCTION

In this paper we describe an adaptive level set approach for computing incompressible two-phase flow in two or three dimensions. Our numerical method is designed for flows characterized by large density and viscosity ratios at the free surface, e.g. air and water,

[1] Support for this work was provided by the Applied Mathematical Sciences Program and the HPCC Grand Challenge Program of the DOE Office of Mathematics, Information, and Computational Sciences under Contract DE-AC03-76SF00098. The first author was also supported in part by Contract DE-FG03-95ER25271 from the U.S. DOE Office of Energy Research Mathematical, Information, and Computational Sciences (MICS) program and an Advanced Scientific Computing Initiative (ASCI) grant from the Los Alamos National Laboratory.

[2] Corresponding author.

and also includes the effects due to surface tension. Our method is also designed for flows in which the free surface separating the two-phases is allowed to merge or break. Incompressible two-phase flow algorithms have been used to model many applications, e.g. drop impact on a pool of water [36], gas bubbles bursting at a free surface [14], ink-jet devices [20], bubbles in a box [21], and water waves [24, 30, 16].

Existing computational methods used to solve incompressible two-phase flow problems include front-tracking methods [50, 21, 49], boundary integral methods [36, 14, 30, 13], volume-of-fluid methods [20, 39, 15], phase field methods [27], capturing methods [47], and level set methods [45, 44, 17, 22].

All of the above methods have their strengths and weaknesses. An advantage of front-tracking methods is that marker particles are introduced explicitly to keep track of the front. This generally reduces by a considerable amount the resolution needed to maintain accuracy comparable to front-capturing methods for the evolution of the free surface. However, regridding algorithms must be employed with front-tracking methods in order to prevent marker particles from coming together at points of large curvature; an explanation of the necessity of regridding is presented in [37]. Another difficulty with front-tracking methods is the fact that extra code needs to be added in order to reconnect for disconnect the free surface separating fluids.

Volume-of-fluid methods are methods based on discretizing the volume fraction of one of the fluids. The motion of the free surface is modeled by solving a conservation law for the volume fraction. As a consequence [39], one can use a conservative finite difference method to update the volume fractions and, except for errors that occur as a result of numerical truncation, the volume of each fluid is conserved. Volume of fluid methods, like level set methods, do not require special procedures to model topological changes of the front. A disadvantage of volume-of-fluid methods is that it is difficult to calculate the curvature of the front from volume fractions.

We shall use the level set approach [45, 37], coupled with incompressible adaptive mesh methodology [3]. Although the level set method does not have the same conservation properties as volume-of-fluid methods or front-tracking methods, the strengths of the level set method lie in its ability to accurately compute flows with surface tension and changes in topology. Furthermore, the level set method generalizes easily to three dimensions. As opposed to front tracking or boundary integral methods, we do not have to add extra code in order to reconnect or disconnect the interface separating fluids. Since we never have to explicitly reconstruct the free surface from the level set function, we avoid complicated front-tracking regridding algorithms or volume-of-fluid reconstruction algorithms. Finally, we use the level set method because the method allows us to accurately compute problems with surface tension. We use the continuum approach [17, 15] in order to represent the surface tension force as a body force. The surface tension term and local interfacial curvature are easily represented in terms of the level set function. In our implementation of the level set method, the level set function will be maintained as the signed distance to the free surface, thus curvature can be accurately computed from the level set function.

We combine the level set approach with the variable density adaptive mesh projection method developed by Almgren *et al.* [3]. Previous adaptive level set implementations have been developed by [32] for computing motion by mean curvature and by [22] for computing thermocapillary motion of deformable drops. We generalize the work of Almgren *et al.* [3] to incompressible two-phase flows in which the density and viscosity ratio between fluids can be 1000:1. Adaptive mesh refinement (AMR) [11, 10] enables us to increase the grid resolution at regions near the free surface and additionally at regions near points of high curvature.

2. GOVERNING EQUATIONS

We use the level set function, ϕ, for tracking the interface between the gas and the liquid [37, 45, 44]. In our algorithm the interface, Γ, is the zero level set of ϕ:

$$\Gamma = \{\boldsymbol{x} \mid \phi(\boldsymbol{x}, t) = 0\}.$$

The level set function ϕ is positive in the liquid and negative in the gas. Hence we have

$$\phi(\boldsymbol{x}, t) \begin{cases} >0, & \text{if } \boldsymbol{x} \in \text{the liquid}, \\ =0, & \text{if } \boldsymbol{x} \in \Gamma, \\ <0, & \text{if } \boldsymbol{x} \in \text{the gas}. \end{cases} \tag{1}$$

The unit normal on the interface, pointing from the gas into the liquid, and the curvature of the interface can easily be expressed in terms of $\phi(\boldsymbol{x}, t)$:

$$\boldsymbol{n} = \left.\frac{\nabla\phi}{|\nabla\phi|}\right|_{\phi=0}, \quad \kappa = \nabla \cdot \left.\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right|_{\phi=0}.$$

Since the interface moves with the fluid, the evolution of ϕ is given by

$$\frac{\partial\phi}{\partial t} + \boldsymbol{u} \cdot \nabla\phi = 0. \tag{2}$$

The governing equation for the fluid velocity and pressure, $\boldsymbol{u}$ and p, along with the free surface boundary conditions can be written as

$$\boldsymbol{U}_t = -\frac{\nabla p}{\rho(\phi)} - (\boldsymbol{U} \cdot \nabla)\boldsymbol{U} + \frac{1}{R}\frac{\nabla \cdot 2\mu\mathcal{D}}{\rho(\phi)} - \frac{1}{W}\frac{\kappa(\phi)\nabla H(\phi)}{\rho(\phi)} + \boldsymbol{F} \tag{3}$$

$$\nabla \cdot \boldsymbol{U} = 0, \tag{4}$$

where $\mathcal{D}$ is the rate of deformation tensor $\mathcal{D} = \frac{1}{2}(\nabla\boldsymbol{U} + \nabla\boldsymbol{U}^{\mathrm{T}})$, $\boldsymbol{F}$ is the gravitational force, ρ and μ are, respectively, the density and viscosity, and $H(\phi)$ is the Heaviside function:

$$H(\phi) = \begin{cases} 0, & \text{if } \phi < 0, \\ \frac{1}{2}, & \text{if } \phi = 0, \\ 1, & \text{if } \phi > 0. \end{cases}$$

The curvature $\kappa(\phi)$ is defined as

$$\kappa(\phi) = \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|}\right).$$

We assume the density and viscosity are constant in each fluid, with values ρ_1 and μ_1, respectively, in the liquid, and ρ_2 and μ_2 in the gas. We define the nondimensionalized quantities

$$\rho(\phi) = H(\phi) + (\rho_2/\rho_1)(1 - H(\phi)) \tag{5}$$

and

$$\mu(\phi) = H(\phi) + (\mu_2/\mu_1)(1 - H(\phi)). \tag{6}$$

The dimensionless parameters used are Reynolds number, $\mathrm{R} = \rho_1 LU/\mu_1$, Froude number, $Fr = U^2/gL$ and Weber number, $W = \rho_1 LU^2/\sigma$.

The Navier–Stokes equations for two-phase flows were written in similar form and used by Unverdi and Tryggvason [50]. The fact that the surface tension can be written as a body force concentrated at the interface has been used by Unverdi and Tryggvason [50] and Brackbill, Kothe, and Zemach [15]. The form we use here is due to Chang, Hou, Merriman, and Osher [17].

2.1. *Projection Methodology*

The method used to solve for velocity and pressure is a variable density approximate projection method described by [5, 39]. We rewrite (3) as

$$\boldsymbol{U}_t + \frac{1}{\rho(\phi)}\nabla p = \boldsymbol{V}(\boldsymbol{U}, \phi). \tag{7}$$

We then take the divergence of both sides of (7) and use the fact that $\nabla \cdot \boldsymbol{U}_t = 0$ in order to reduce (3) and (4) into a single equation for pressure,

$$\nabla \cdot \frac{1}{\rho}\nabla p = \nabla \cdot \boldsymbol{V}. \tag{8}$$

After solving (8) for ∇p the updated value for $\boldsymbol{U}_t$ is

$$\boldsymbol{U}_t = \boldsymbol{V} - \nabla p/\rho. \tag{9}$$

For future reference, we define the projection operator $\boldsymbol{P}$ as

$$\boldsymbol{U}_t \equiv \boldsymbol{P}(\boldsymbol{V}). \tag{10}$$

Combining (10) and (9) yields

$$\nabla p/\rho = \boldsymbol{V} - \boldsymbol{U}_t \equiv \boldsymbol{V} - \boldsymbol{P}(\boldsymbol{V}) \equiv (I - P)(\boldsymbol{V}). \tag{11}$$

3. SINGLE-GRID DISCRETIZATION

Our single grid discretization procedure for approximating (2) and (3) is based on the variable density projection method described by Bell *et al.* [7], Bell and Marcus [9], Almgren *et al.* [5], and Puckett *et al.* [39]. For the single grid discretization, we have uniform grid spacing $\Delta x = \Delta y = h$. The discrete velocity field $\boldsymbol{U}^n_{i,j,k}$ and the discrete level set function $\phi^n_{i,j,k}$ are located at cell centers. The pressure $p^{n-1/2}_{i+1/2,j+1/2,k+1/2}$ is located at cell corners. A diagram of where the discrete variables are located in relation to the computational grid is shown in Fig. 1. J represents the index of the computational cell closest to the top physical boundary.

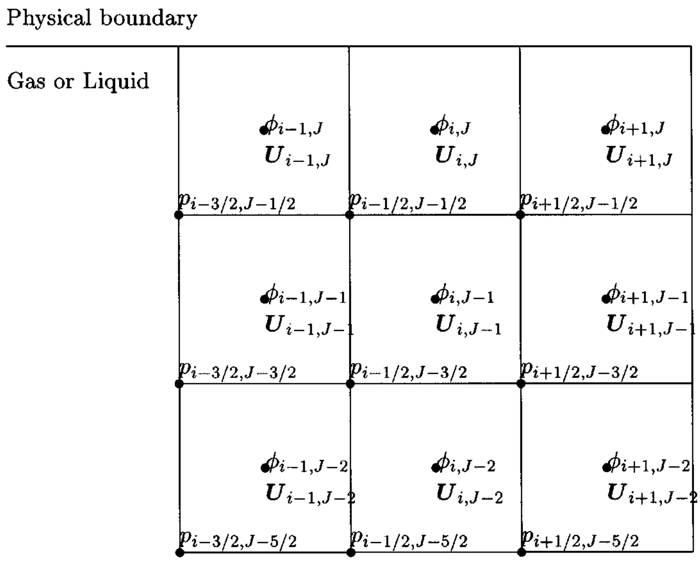
Physical boundary
Gas or Liquid
$\phi_{i-1,J}$ $U_{i-1,J}$
$\phi_{i,J}$ $U_{i,J}$
$\phi_{i+1,J}$ $U_{i+1,J}$
$p_{i-3/2,J-1/2}$ $p_{i-1/2,J-1/2}$ $p_{i+1/2,J-1/2}$
$\phi_{i-1,J-1}$ $U_{i-1,J-1}$
$\phi_{i,J-1}$ $U_{i,J-1}$
$\phi_{i+1,J-1}$ $U_{i+1,J-1}$
$p_{i-3/2,J-3/2}$ $p_{i-1/2,J-3/2}$ $p_{i+1/2,J-3/2}$
$\phi_{i-1,J-2}$ $U_{i-1,J-2}$
$\phi_{i,J-2}$ $U_{i,J-2}$
$\phi_{i+1,J-2}$ $U_{i+1,J-2}$
$p_{i-3/2,J-5/2}$ $p_{i-1/2,J-5/2}$ $p_{i+1/2,J-5/2}$

FIG. 1. Diagram of where the discrete variables $\boldsymbol{U}$, p, and ϕ are located in relation to the computational grid and the physical boundary.

3.1. *Temporal Discretization*

The time-stepping procedure is based on the Crank–Nicholson method. At the beginning of each time step, we are given the velocity $\boldsymbol{U}^n$ and the level set function ϕ^n at time t^n. We are also given the lagged pressure gradient $p^{n-1/2}$. The density $\rho^n = \rho(\phi^n)$, viscosity $\mu^n = \mu(\phi^n)$ and Heaviside function $H^n = H(\phi^n)$ are given at time t^n since they are functions of ϕ^n. We discretize (3) and (2) in time using the following steps:

1. Level set update for ϕ^{n+1}:

$$\phi^{n+1} = \phi^n - \Delta t[\boldsymbol{U} \cdot \nabla\phi]^{n+1/2}. \tag{12}$$

Here, the brackets $[\cdots]^{n+1/2}$ mean the discrete version of the continuous operator. The nonlinear advection term $[\boldsymbol{U} \cdot \nabla\phi]^{n+1/2}$ is evaluated using an explicit predictor–corrector scheme and requires only the available data at t^n. In Section 3.2, we give a description of how $[\boldsymbol{U} \cdot \nabla\phi]^{n+1/2}$ is formed. Once ϕ^{n+1} is obtained from (12), the following quantities are updated:

$$\phi^{n+1/2} = \frac{1}{2}(\phi^n + \phi^{n+1}) \tag{13}$$

$$\rho^{n+1/2} = \rho(\phi^{n+1/2}) \tag{14}$$

$$\mu^{n+1/2} = \mu(\phi^{n+1/2}). \tag{15}$$

2. Semi-implicit viscous solve for the intermediate velocity $\boldsymbol{U}^*$:

$$\frac{\boldsymbol{U}^* - \boldsymbol{U}^n}{\Delta t} = -[(\boldsymbol{U} \cdot \nabla)\boldsymbol{U}]^{n+1/2} - \frac{Gp^{n-1/2}}{\rho^{n+1/2}} + \frac{\mathcal{L}^* + \mathcal{L}^n}{2\rho^{n+1/2}} - \frac{\mathcal{M}^{n+1/2}}{\rho^{n+1/2}} + \boldsymbol{F}. \tag{16}$$

$\mathcal{L}$ is a second-order finite difference approximation to $(1/\mathrm{R})\nabla\cdot(2\mu(\phi)\mathcal{D})$, $\mathcal{M}$ is a finite difference approximation to $(1/\mathrm{W})\kappa(\phi)\nabla H$, and Gp is an approximation to ∇p. In Section 3.3 we give a description of Gp, $\mathcal{L}$, and $\mathcal{M}$. The nonlinear advection term $[(\boldsymbol{U}\cdot\nabla)\boldsymbol{U}]^{n+1/2}$ is evaluated using an explicit predictor–corrector scheme and requires only the available data at t^n. In Section 3.2, we give a description of how $[(\boldsymbol{U}\cdot\nabla)\boldsymbol{U}]^{n+1/2}$ is discretized. The density ρ, viscosity μ, Heaviside function H, and curvature κ are constructed from the level set function calculated at time $t^{n+1/2}$ in the level set advection step (13). The lagged pressure gradient $Gp^{n-1/2}$ and force $\boldsymbol{F}$ are treated as source terms. Equation (16) when discretized results in a coupled parabolic solve for all velocity components of $\boldsymbol{U}^*$. We use multigrid as an iteration method for solving (16).

3. Projection step for $\boldsymbol{U}^{n+1}$:

$$\begin{aligned}\frac{\boldsymbol{U}^{n+1}-\boldsymbol{U}^n}{\Delta t} &= \boldsymbol{\mathcal{P}}\left(\frac{\boldsymbol{U}^*-\boldsymbol{U}^n}{\Delta t}\right)\\ \frac{1}{\rho^{n+1/2}}Gp^{n+1/2} &= \frac{1}{\rho^{n+1/2}}Gp^{n-1/2}+(\boldsymbol{I}-\boldsymbol{\mathcal{P}})\left(\frac{\boldsymbol{U}^*-\boldsymbol{U}^n}{\Delta t}\right).\end{aligned} \tag{17}$$

$\boldsymbol{\mathcal{P}}$ represents the discretization of the projection operator (10). In Section 3.4 we give a description of $\boldsymbol{\mathcal{P}}$.

4. Redistance step for ϕ^{n+1}. We maintain the level set function ϕ as the signed normal distance to the free surface. In Section 3.5, we give a description of the redistance step.

3.1.1. Timestep. The timestep Δt at time t^n is determined by restrictions due to the CFL condition, gravity, viscosity and surface tension [45, 15]:

$$\Delta t < \min_{i,j,k}\left(\sqrt{W\frac{(\rho_1+\rho_2)}{8\pi}}\Delta x^{3/2}, \frac{3}{14}R\frac{\rho^n\Delta x^2}{\mu^n}, \frac{\Delta x}{|\boldsymbol{u}^n|}, \sqrt{\frac{2\Delta x}{\mathcal{F}^n}}\right),$$

where

$$\mathcal{F}^n = \left|-\frac{Gp^{n-1/2}}{\rho^n}+\frac{\mathcal{L}^n}{\rho^n}-\frac{\mathcal{M}^n}{\rho^n}+\boldsymbol{F}\right|.$$

We note that, even though we handle the viscous terms semi-implicitly, we have still found a need for the stringent timestep constraint. One reason for this, as pointed out by Almgren *et al.* [3] and Minion [34], is the fact that viscous terms are not included in defining the states used in the transverse derivatives. Here, since our flows are not dominated by viscous effects, we choose to handle the viscous terms semi-implicitly in order to preserve a high order of accuracy. In Section 5.2, we found that the percentage of time spent solving (16) was 16%. In [46], flows involving oil and water were computed; since the viscosity of oil is very large in comparison to water, the viscous terms were handled implicitly by [46] and no viscous timestep restriction was necessary.

3.2. *Approximation of the Advection Terms*

In this section, we describe the discretization of the advection terms

$$[(\boldsymbol{U}\cdot\nabla)\boldsymbol{U}]^{n+1/2} \tag{18}$$

and

$$[\boldsymbol{U}\cdot\nabla\phi]^{n+1/2}. \tag{19}$$

The discretization of the advection terms in this algorithm is very similar to the discretization used by [39, 3]. It is a predictor–corrector method based on the unsplit Godunov method introduced by Colella [19].

In the predictor we extrapolate the velocity $\boldsymbol{U}$ and the level set function ϕ to the cell faces at $t^{n+1/2}$ using a second-order Taylor series expansion in space and time. The time derivative $\boldsymbol{U}_t$ is replaced using (3) and the time derivative ϕ_t is replaced using (2). For face $(i+1/2, j, k)$ this gives

$$\begin{aligned}\boldsymbol{U}^{n+1/2,L}_{i+1/2,j,k} &= \boldsymbol{U}^n_{ijk} + \left(\frac{\Delta x}{2} - \frac{u^n_{ijk}\Delta t}{2}\right)\boldsymbol{U}^n_{x,ijk} - \frac{\Delta t}{2}(\widehat{v\boldsymbol{U}_y})_{ijk} - \frac{\Delta t}{2}(\widehat{w\boldsymbol{U}_z})_{ijk}\\ &\quad + \frac{\Delta t}{2}\left(-\frac{Gp^{n-1/2}_{ijk}}{\rho^n_{ijk}} + \frac{\mathcal{L}^n_{ijk}}{\rho^n_{ijk}} - \frac{\mathcal{M}^n_{ijk}}{\rho^n_{ijk}} + \boldsymbol{F}\right) \qquad (20)\end{aligned}$$

$$\phi^{n+1/2,L}_{i+1/2,j,k} = \phi^n_{ijk} + \left(\frac{\Delta x}{2} - \frac{u^n_{ijk}\Delta t}{2}\right)\phi^n_{x,ijk} - \frac{\Delta t}{2}(\widehat{v\phi_y})_{ijk} - \frac{\Delta t}{2}(\widehat{w\phi_z})_{ijk}, \qquad (21)$$

extrapolated from cell (i, j, k), and

$$\begin{aligned}\boldsymbol{U}^{n+1/2,R}_{i+1/2,j,k} &= \boldsymbol{U}^n_{i+1,jk} - \left(\frac{\Delta x}{2} + \frac{u^n_{i+1,jk}\Delta t}{2}\right)\boldsymbol{U}^n_{x,i+1,jk} - \frac{\Delta t}{2}(\widehat{v\boldsymbol{U}_y})_{i+1,jk} - \frac{\Delta t}{2}(\widehat{w\boldsymbol{U}_z})_{i+1,jk}\\ &\quad + \frac{\Delta t}{2}\left(-\frac{Gp^{n-1/2}_{i+1,jk}}{\rho^n_{i+1,jk}} + \frac{\mathcal{L}^n_{i+1,jk}}{\rho^n_{i+1,jk}} - \frac{\mathcal{M}^n_{i+1,jk}}{\rho^n_{i+1,jk}} + \boldsymbol{F}\right) \qquad (22)\end{aligned}$$

$$\phi^{n+1/2,R}_{i+1/2,j,k} = \phi^n_{i+1,jk} - \left(\frac{\Delta x}{2} + \frac{u^n_{i+1,jk}\Delta t}{2}\right)\phi^n_{x,i+1,jk} - \frac{\Delta t}{2}(\widehat{v\phi_y})_{i+1,jk} - \frac{\Delta t}{2}(\widehat{w\phi_z})_{i+1,jk}, \qquad (23)$$

extrapolated from cell $(i+1, j, k)$.

Analogous formulae are used to predict values at each of the other faces of the cell:

$$\boldsymbol{U}^{n+1/2,F/B}_{i,j+1/2,k}, \boldsymbol{U}^{n+1/2,U/D}_{ij,k+1/2}, \phi^{n+1/2,F/B}_{i,j+1/2,k}, \phi^{n+1/2,U/D}_{ij,k+1/2}. \qquad (24)$$

The first derivatives normal to the face, $\boldsymbol{U}^n_x$, and ϕ^n_x for the example in (20) and (23) are evaluated using a monotonicity-limited fourth-order slope approximation [18]. The limiting is done on each component of the velocity at t^n individually.

The transverse derivative terms,

$$\widehat{v\boldsymbol{U}_y}, \widehat{w\boldsymbol{U}_z}, \widehat{v\phi_y}, \widehat{w\phi_z},$$

are evaluated by first extrapolating $\boldsymbol{U}$ and ϕ to the transverse faces from the cell centers on either side, using normal derivatives only, and then choosing between these states using the upwinding procedure as described in detail by Almgren *et al.* [3] and Puckett *et al.* [39].

Once we have computed $u^{n+1/2,L/R}_{i+1/2,jk}$, $v^{n+1/2,F/B}_{i,j+1/2,k}$, and $w^{n+1/2,T/B}_{ij,k+1/2}$, we are in a position to construct the normal face-centered edge velocities at $t^{n+1/2}$:

$$u^{ADV}_{i+1/2,jk}, v^{ADV}_{i,j+1/2,k}, w^{ADV}_{ij,k+1/2}.$$

Given $u^{n+1/2,L}_{i+1/2,jk}$ and $u^{n+1/2,R}_{i+1/2,jk}$, we use an upwinding procedure to choose $u^{n+1/2}_{i+1/2,jk}$:

$$u^{n+1/2}_{i+1/2,jk} = \begin{cases} u^L, & \text{if } u^L > 0;\ u^L + u^R > 0, \\ 0, & \text{if } u^L \le 0,\ u^R \ge 0, \text{ or } u^L + u^R = 0, \\ u^R & \text{if } u^R < 0;\ u^L + u^R < 0. \end{cases} \tag{25}$$

Here, we suppress the $i + 1/2,\ j, k$ spatial indices on left and right states and we also suppress the $n + 1/2$ temporal index.

We follow a similar procedure as in (25) to construct $v^{n+1/2}_{i,j+1/2,k}$ and $w^{n+1/2}_{i,j,k+1/2}$.

These normal velocities on cell faces at $t^{n+1/2}$,

$$u^{n+1/2}_{i+1/2,jk},\ v^{n+1/2}_{i,j+1/2,k},\ w^{n+1/2}_{ij,k+1/2}, \tag{26}$$

are second-order accurate but do not, in general, satisfy the discrete divergence-free condition. In order to make these velocities divergence-free, we apply the MAC projection [8]. The equation

$$D^{\mathrm{MAC}}\left(\frac{1}{\rho^n} G^{\mathrm{MAC}} p^{\mathrm{MAC}}\right) = D^{\mathrm{MAC}}(\boldsymbol{U}^{n+1/2}) \tag{27}$$

is solved for p^{MAC}, where

$$D^{\mathrm{MAC}}\boldsymbol{U}^{n+1/2} = \frac{u^{n+1/2}_{i+1/2,j,k} - u^{n+1/2}_{i-1/2,j,k}}{\Delta x} + \frac{v^{n+1/2}_{i,j+1/2,k} - v^{n+1/2}_{i,j-1/2,k}}{\Delta y} + \frac{w^{n+1/2}_{i,j,k+1/2} - w^{n+1/2}_{i,j,k-1/2}}{\Delta z}$$

and $G^{\mathrm{MAC}} = -(D^{\mathrm{MAC}})^{\mathrm{T}}$ so that

$$\left(G^{\mathrm{MAC}}_x p^{\mathrm{MAC}}\right)_{i+1/2,j,k} = \frac{\left(p^{\mathrm{MAC}}_{i+1,j,k} - p^{\mathrm{MAC}}_{i,j,k}\right)}{\Delta x}$$

with G^{MAC}_y and G^{MAC}_z defined analogously. The resulting linear system (27) is solved using a multigrid preconditioned conjugate gradient solver [48].

The face-based advection velocities at $t^{n+1/2}$ are then defined by

$$u^{\mathrm{ADV}}_{i+1/2,j,k} = u^{n+1/2}_{i+1/2,j,k} - \frac{1}{\rho^n_{i+1/2,j,k}}\left(G^{\mathrm{MAC}}_x p^{\mathrm{MAC}}\right)_{i+1/2,j,k} \tag{28}$$

with $v^{\mathrm{ADV}}_{i,j+1/2,k}$ and $w^{\mathrm{ADV}}_{i,j,k+1/2}$ defined analogously. The quantity $\rho^n_{i+1/2,j,k}$ in (28) is defined by

$$\rho^n_{i+1/2,j,k} = \frac{1}{2}\left(\rho^n_{ijk} + \rho^n_{i+1,jk}\right)$$

with $\rho^n_{i,j+1/2,k}$ and $\rho^n_{i,j,k+1/2}$ defined analogously.

The next step, after constructing the advective velocities

$$u^{\mathrm{ADV}}_{i+1/2,j,k},\ v^{\mathrm{ADV}}_{i,j+1/2,k},\ w^{\mathrm{ADV}}_{i,j,k+1/2},$$

is to choose the appropriate states $\boldsymbol{U}^{n+1/2}_{i+1/2,jk}, \phi^{n+1/2}_{i+1/2,jk}$ given the left and right states in (20) thru (23):

$$\boldsymbol{U}^{n+1/2,L}_{i+1/2,j,k}, \boldsymbol{U}^{n+1/2,R}_{i+1/2,j,k}, \phi^{n+1/2,L}_{i+1/2,j,k}, \phi^{n+1/2,R}_{i+1/2,j,k}.$$

We have

$$\boldsymbol{U}^{n+1/2}_{i+1/2,jk} = \begin{cases} \boldsymbol{U}^L, & \text{if } u^{\text{ADV}} > 0, \\ \frac{1}{2}(\boldsymbol{U}^L + \boldsymbol{U}^R), & \text{if } u^{\text{ADV}} = 0, \\ \boldsymbol{U}^R, & \text{if } u^{\text{ADV}} < 0; \end{cases} \tag{29}$$

$$\phi^{n+1/2}_{i+1/2,jk} = \begin{cases} \phi^L, & \text{if } u^{\text{ADV}} > 0, \\ \frac{1}{2}(\phi^L + \phi^R), & \text{if } u^{\text{ADV}} = 0, \\ \phi^R, & \text{if } u^{\text{ADV}} < 0. \end{cases} \tag{30}$$

Here, we suppress the $i + 1/2,\ j, k$ spatial indices on left and right states and we also suppress the $n + 1/2$ temporal index.

We follow a similar procedure as in (30) and (29) to construct

$$\boldsymbol{U}^{n+1/2}_{i,j+1/2,k}, \boldsymbol{U}^{n+1/2}_{i,j,k+1/2}, \phi^{n+1/2}_{i,j+1/2,k}, \phi^{n+1/2}_{i,j,k+1/2}.$$

The advection terms can now be defined by

$$\begin{aligned} [(\boldsymbol{U}\cdot\nabla)\boldsymbol{U}]^{n+1/2}_{i,j,k} &= \frac{1}{\Delta x}\frac{u^{\text{ADV}}_{i+1/2,j,k} + u^{\text{ADV}}_{i-1/2,j,k}}{2}(\boldsymbol{U}_{i+1/2,j,k} - \boldsymbol{U}_{i-1/2,j,k}) \\ &\quad + \frac{1}{\Delta y}\frac{v^{\text{ADV}}_{i,j+1/2,k} + v^{\text{ADV}}_{i,j-1/2,k}}{2}(\boldsymbol{U}_{i,j+1/2,k} - \boldsymbol{U}_{i,j-1/2,k}) \\ &\quad + \frac{1}{\Delta z}\frac{w^{\text{ADV}}_{i,j,k+1/2} + w^{\text{ADV}}_{i,j,k-1/2}}{2}(\boldsymbol{U}_{i,j,k+1/2} - \boldsymbol{U}_{i,j,k-1/2}); \end{aligned} \tag{31}$$

$$\begin{aligned} [(\boldsymbol{U}\cdot\nabla)\phi]^{n+1/2}_{i,j,k} &= \frac{1}{\Delta x}\frac{u^{\text{ADV}}_{i+1/2,j,k} + u^{\text{ADV}}_{i-1/2,j,k}}{2}(\phi_{i+1/2,j,k} - \phi_{i-1/2,j,k}) \\ &\quad + \frac{1}{\Delta y}\frac{v^{\text{ADV}}_{i,j+1/2,k} + v^{\text{ADV}}_{i,j-1/2,k}}{2}(\phi_{i,j+1/2,k} - \phi_{i,j-1/2,k}) \\ &\quad + \frac{1}{\Delta z}\frac{w^{\text{ADV}}_{i,j,k+1/2} + w^{\text{ADV}}_{i,j,k-1/2}}{2}(\phi_{i,j,k+1/2} - \phi_{i,j,k-1/2}). \end{aligned} \tag{32}$$

3.3. *Discretization of Pressure Gradient, Viscous and Surface Tension Terms*

In this section we describe the finite difference approximation in two dimensions to the pressure gradient, Gp, viscous term, $\mathcal{L}$, and surface tension term, $\mathcal{M}$. The finite difference approximations in three dimensions follow analogously.

The discrete pressure gradient is defined by

$$(Gp)_{ij} \equiv \begin{pmatrix} \frac{p_{i+1/2,j+1/2} + p_{i+1/2,j-1/2} - p_{i-1/2,j+1/2} - p_{i-1/2,j-1/2}}{2\Delta x} \\ \frac{p_{i+1/2,j+1/2} - p_{i+1/2,j-1/2} + p_{i-1/2,j+1/2} - p_{i-1/2,j-1/2}}{2\Delta y} \end{pmatrix}, \tag{33}$$

where G here denotes a discrete gradient operator defined at cell centers but operating on nodal data.

The first component of the viscous term $(1/\mathrm{R})\nabla \cdot 2\mu(\phi)\mathcal{D}$ is discretized as

$$(\mathcal{L})^1_{ij} = \frac{1}{\mathrm{R}}\left(\begin{array}{l} \frac{2\mu_{i+1/2,j}(u_{i+1,j}-u_{i,j}) - 2\mu_{i-1/2,j}(u_{i,j}-u_{i-1,j})}{\Delta x^2} \\ \quad + \frac{\mu_{i,j+1/2}(u_{i,j+1}-u_{i,j}) - \mu_{i,j-1/2}(u_{i,j}-u_{i,j-1})}{\Delta y^2} \\ \quad + \frac{\mu_{i,j+1/2}(v_{i+1,j+1}-v_{i-1,j+1}+v_{i+1,j}-v_{i-1,j}) - \mu_{i,j-1/2}(v_{i+1,j}-v_{i-1,j}+v_{i+1,j-1}-v_{i-1,j-1})}{\Delta x \Delta y} \end{array}\right),$$

where

$$\mu_{i+1/2,j} = \frac{1}{2}(\mu(\phi_{i,j}) + \mu(\phi_{i+1,j})), \quad \mu_{i,j+1/2} = \frac{1}{2}(\mu(\phi_{i,j}) + \mu(\phi_{i,j+1})).$$

The second component of the viscous term, $(\mathcal{L})^2_{ij}$, is discretized in a similar manner.

The surface tension term $(1/\mathrm{W})\kappa(\phi)\nabla H(\phi)$ is discretized as

$$(\mathcal{M})_{ij} \equiv \frac{1}{W}(D\boldsymbol{N})_{ij}(GH^{\text{node}})_{ij}. \tag{34}$$

$\boldsymbol{N}_{i+1/2,j+1/2}$ is the discrete approximation of the level set normal $\nabla\phi/|\nabla\phi|$,

$$\boldsymbol{N}_{i+1/2,j+1/2} \equiv \frac{(G\phi)_{i+1/2,j+1/2}}{|(G\phi)_{i+1/2,j+1/2}|}, \tag{35}$$

where

$$(G\phi)_{i+1/2,j+1/2} \equiv \begin{pmatrix} \frac{\phi_{i+1,j+1}+\phi_{i+1,j}-\phi_{i,j+1}-\phi_{i,j}}{2\Delta x} \\ \frac{\phi_{i+1,j+1}-\phi_{i+1,j}+\phi_{i,j+1}-\phi_{i,j}}{2\Delta y} \end{pmatrix}. \tag{36}$$

Here we use G to refer to the discrete gradient operator defined on nodes but operating on cell-centered data.

We define the cell-based discrete divergence operator D by

$$(D\boldsymbol{N})_{ij} \equiv \frac{n^1_{i+1/2,j-1/2} + n^1_{i+1/2,j+1/2} - n^1_{i-1/2,j-1/2} - n^1_{i-1/2,j+1/2}}{\Delta x} \tag{37}$$

$$+ \frac{n^2_{i+1/2,j+1/2} - n^2_{i+1/2,j-1/2} - n^2_{i-1/2,j-1/2} + n^2_{i-1/2,j+1/2}}{\Delta y}. \tag{38}$$

The node-based Heaviside function $H^{\text{node}}_{i+1/2,j+1/2}$ is defined as the average of the four surrounding cell-based Heaviside functions:

$$H^{\text{node}}_{i+1/2,j+1/2} = \frac{H(\phi_{i+1,j}) + H(\phi_{i,j}) + H(\phi_{i+1,j+1}) + H(\phi_{i,j+1})}{4}. \tag{39}$$

3.4. *Discretization of the Projection*

In this section we describe the discrete "approximate projection," $\boldsymbol{\mathcal{P}}$, which is used in (17). $\boldsymbol{\mathcal{P}}$ is an approximation to the projection operator $\boldsymbol{P}$ described in (10). We remark that a detailed description of the approximate projection is given by [5].

Given the discrete vector field

$$\frac{\boldsymbol{U}^* - \boldsymbol{U}^n}{\Delta t}, \tag{40}$$

we decompose (40) into an *approximately* divergence-free part

$$\frac{\boldsymbol{U}^{n+1} - \boldsymbol{U}^n}{\Delta t} \tag{41}$$

and the discrete gradient of a scalar q divided by density

$$\frac{(Gq)_{ij}}{\rho_{ij}^{n+1/2}}, \tag{42}$$

where the discrete gradient G in (42) is defined in (33).

The approximate projection is computed by solving

$$L_\rho q = D\left(\frac{\boldsymbol{U}^* - \boldsymbol{U}^n}{\Delta t}\right) \tag{43}$$

for q. The right-hand side of (43) is an approximation to $\nabla \cdot \boldsymbol{V}$ found in the right-hand side of (8). The discrete divergence $D\boldsymbol{U}$ is

$$\begin{aligned}(D\boldsymbol{U})_{i+1/2,j+1/2} = {} & \frac{u_{i+1,j} + u_{i+1,j+1} - u_{i,j} - u_{i,j+1}}{\Delta x} \\ & + \frac{v_{i+1,j+1} - v_{i+1,j} - v_{i,j} + v_{i,j+1}}{\Delta y}.\end{aligned} \tag{44}$$

The left-hand side of (43), $L_\rho q$, is an approximation to $\nabla \cdot (1/\rho)\nabla p$ found in the left-hand side of (8). The discrete representation of $L_\rho q$ is

$$\begin{aligned}&(L_\rho q)_{i+1/2,j+1/2} \\ &= \frac{1}{6h^2}\left(\begin{aligned}&\tfrac{1}{\rho_{i,j}}(2q_{i-1/2,j-1/2} + q_{i+1/2,j-1/2} + q_{i-1/2,j+1/2} - 4q_{i+1/2,j+1/2}) \\ &\quad + \tfrac{1}{\rho_{i,j+1}}(2q_{i-1/2,j+3/2} + q_{i+1/2,j+3/2} + q_{i-1/2,j+1/2} - 4q_{i+1/2,j+1/2}) \\ &\quad + \tfrac{1}{\rho_{i+1,j}}(2q_{i+3/2,j-1/2} + q_{i+1/2,j-1/2} + q_{i+3/2,j+1/2} - 4q_{i+1/2,j+1/2}) \\ &\quad + \tfrac{1}{\rho_{i+1,j+1}}(2q_{i+3/2,j+3/2} + q_{i+1/2,j+3/2} + q_{i+3/2,j+1/2} - 4q_{i+1/2,j+1/2})\end{aligned}\right).\end{aligned} \tag{45}$$

In two dimensions the operator $L_\rho q$ (45) is derived from the variational form of (8),

$$\int \frac{1}{\rho}\nabla q(\mathbf{x}) \cdot \nabla\psi(\mathbf{x})\, d\mathbf{x} = \int \frac{\boldsymbol{U}^* - \boldsymbol{U}^n}{\Delta t} \cdot \nabla\psi(\mathbf{x})\, d\mathbf{x} \quad \forall\psi(\mathbf{x}), \tag{46}$$

where $d\mathbf{x}$ is the volume element $dx\,dy, r\,dr\,d\theta,$ or $dx\,dy\,dz,$ as appropriate. The finite element basis functions $\psi(\mathbf{x})$ represent standard piecewise bilinear functions. In three dimensions we use a standard seven-point approximation in order to derive $L_\rho q$.

After (43) is solved, we form $(\boldsymbol{U}^{n+1} - \boldsymbol{U}^n)/\Delta t$,

$$\frac{\boldsymbol{U}^{n+1} - \boldsymbol{U}^n}{\Delta t} = \frac{\boldsymbol{U}^* - \boldsymbol{U}^n}{\Delta t} - \frac{Gq}{\rho^{n+1/2}}, \tag{47}$$

and $p^{n+1/2}$,

$$p^{n+1/2} = p^{n-1/2} + q.$$

Remarks. • The discrete projection step presented here is slightly different from the continuous analogue presented in Section 2.1 because we are solving for the *difference* in pressure $q = p^{n+1/2} - p^{n-1/2}$, instead of the actual pressure $p^{n+1/2}$.

• The discrete projection operator $\mathcal{P}$ is called an approximate projection because the discrete divergence of (41),

$$\left[D\left(\frac{\boldsymbol{U}^{n+1} - \boldsymbol{U}^n}{\Delta t}\right)\right]_{i+1/2,j+1/2}, \tag{48}$$

is not identically zero. In order to see why (48) is not necessarily zero, we apply the discrete divergence D to both sides of (47) in order to arrive at

$$\left[D\left(\frac{\boldsymbol{U}^{n+1} - \boldsymbol{U}^n}{\Delta t}\right)\right]_{i+1/2,j+1/2} = \left[D\left(\frac{\boldsymbol{U}^* - \boldsymbol{U}^n}{\Delta t}\right)\right]_{i+1/2,j+1/2} - \left[D\frac{1}{\rho^{n+1/2}}Gq\right]_{i+1/2,j+1/2}. \tag{49}$$

The discrete operator $D(1/\rho^{n+1/2})Gq$ is not the same as $L_\rho q$, which means (48) is not necessarily zero.

3.4.1. Matrix solver. In order to compute the approximate projection $\mathcal{P}$, we solve (43) for $q_{i+1/2,j+1/2}$. We use the multigrid-preconditioned conjugate gradient method (MGPCG) [48] for solving (43).

The boundary conditions for (43) are homogeneous Neumann at a solid wall or at an axis of symmetry. The boundary conditions at outflow boundaries are homogeneous Dirichlet.

In preliminary development, we attempted to use standard multigrid techniques to solve (43). These standard multigrid techniques used the coefficients $1/\rho$ in defining the interpolation operator [3, 1], but would not converge for many problems with high density ratios. As an example, for an axisymmetric bubble rise problem with no surface tension, standard multigrid took an order of magnitude more iterations than MGPCG at the point near bubble breakup.

As a result, we have implemented the multigrid-preconditioned conjugate gradient method [48] to solve (43). This allows us to run the bubble and drop problems that previously failed at the proper density ratio (816:1).

The preconditioner is a single multigrid V-cycle [51] with the following properties, motivated by the need for the preconditioner to a conjugate gradient solve to be symmetric:

• The interpolation and restriction operators have no coefficient-weighting and satisfy $cR^T = I$; R refers to the restriction operator and I refers to the interpolation operator.

• Symmetric multicolor Gauss–Seidel relaxation is used as the smoother at each level of the V-cycle. For the nine-point stencil in two dimensions, we use a four-color Gauss–Seidel relaxation step. On the way down the V-cycle, the order is RBGW. On the way up, the order is WGBR. Likewise, for the three-dimensional seven-point stencil, we use a multicolored relaxation scheme in which the ordering is again reversed on the way up the V-cycle.

• At the coarsest level of the V-cycle, the "bottom solver" is a preconditioned conjugate gradient solver. The preconditioner for this bottom solver is again symmetric multicolor Gauss–Seidel relaxation as described above; i.e. RBGW on the way down the V-cycle, and WGBR on the way up. The equation at the coarsest level must be solved to a tolerance two orders of magnitude smaller than the tolerance of the overall conjugate gradient solver, or the multigrid as preconditioner will not be sufficiently symmetric.

• If the boundary conditions are all homogeneous Neumann, discrete solvability is enforced by ensuring that the sum of the right-hand side of (43) is zero.

• The elliptic operator at each level of the V-cycle is identical in form but with coarsened coefficients from the finer levels. The coefficients $1/\rho$ are each associated with a directional flux and are coarsened by doing an arithmetic average transverse to each "flux" and a harmonic average parallel to the flux; we refer the reader to [3] for details of this procedure.

3.5. *Interface Thickness*

We shall give the interface a thickness as was done in the work of [50, 45]. Numerically, we substitute the smoothed Heaviside function $H_\epsilon(\phi)$ for the sharp Heaviside function $H(\phi)$. The smoothed Heaviside function is defined as

$$H_\varepsilon(\phi) = \begin{cases} 0, & \text{if } \phi < -\varepsilon, \\ \frac{1}{2}\left[1 + \frac{\phi}{\varepsilon} + \frac{1}{\pi}\sin(\pi\phi/\varepsilon)\right], & \text{if } |\phi| \le \varepsilon, \\ 1, & \text{if } \phi > \varepsilon. \end{cases} \tag{50}$$

Assume that ϕ represents the signed normal distance to the free surface. The $1/2$ contour of the sharp Heaviside function $H(\phi)$ will show up on a contour plot with jagged or staircase contours. Although ϕ is smooth, $H(\phi)$ has a jump at the zero levelset. A contour plot of $H_\varepsilon(\phi)$ where $\varepsilon = \alpha\Delta x$ will not show up having a jagged shape when $\alpha > 1$. By giving the interface a thickness of 2ε we eliminate problems when solving (43) and also when discretizing the surface tension term

$$\frac{1}{\mathrm{W}}\frac{\kappa(\phi)\nabla H(\phi)}{\rho(\phi)}.$$

In our algorithm, the front will have a uniform thickness which means we require that the level set function ϕ represent the signed normal distance to the free surface; in other words, ϕ is a distance function. It is clear that we can choose $\phi(\boldsymbol{x}, 0)$ to be a distance function; however, under the evolution of (2) it will not necessarily remain one.

In order to maintain $\phi(\boldsymbol{x}, t)$ as a distance function, we must be able to solve the following problem: given a level set function $\phi(\boldsymbol{x}, t)$, reinitialize it so that it is a distance function for $|\phi| < \varepsilon$ *without changing its zero level set*. This is achieved (see [45, 43]) by performing the following steps:

1. $d(\boldsymbol{x}, 0) = \phi(\boldsymbol{x}, t)$.

2. For $\tau = 0 \cdots \varepsilon$ solve

$$\frac{\partial d}{\partial \tau} = S(\phi)(1 - |\nabla d|), \tag{51}$$

where

$$S(\phi) = \begin{cases} -1, & \text{if } \phi < 0, \\ 0, & \text{if } \phi = 0, \\ 1, & \text{if } \phi > 0, \end{cases} \tag{52}$$

and τ is an artificial time.

3. $\phi(\boldsymbol{x}, t) = d(\boldsymbol{x}, \varepsilon)$.

The steady solutions of (51) are distance functions. Furthermore, since $S(0) = 0$, then $d(\boldsymbol{x}, \tau)$ has the same zero level set as $\phi(\boldsymbol{x}, t)$.

We only need to solve (51) for $\tau = 0 \cdots \varepsilon$ because the level set function is reinitialized near the front first. To see this we rewrite (51) as

$$d_\tau + \boldsymbol{w} \cdot \nabla d = S(\phi), \tag{53}$$

where

$$\boldsymbol{w} = S(\phi)\frac{\nabla d}{|\nabla d|}.$$

It is evident that (53) is a nonlinear hyperbolic equation with the characteristic velocities pointing *outwards* from the interface in the direction of the normal. This means that d will be reinitialized to $|\nabla d| = 1$ near the interface first. Since we only need the level set function to be a distance function near the interface, it is only necessary to solve (53) for $\tau = 0 \cdots \varepsilon$.

The time-stepping procedure for the redistance equation (51) is based on the second-order Runge–Kutta method. At the beginning of each iteration we are given d^k at "time" τ^k. We then have

$$d^{k,(1)} = d^k + \Delta\tau L(d^k), \tag{54}$$

$$d^{k+1} = d^{k,(1)} + \frac{\Delta\tau}{2}\left(L(d^k) + L\left(d^{k,(1)}\right)\right). \tag{55}$$

$\Delta\tau$ is chosen to be $\Delta x/2$; $\Delta x/2$ satisfies the CFL condition for (53) since $|\boldsymbol{w}| \le 1$. $L(d)$ represents the discretization of the spatial term $S(\phi)(1 - |\nabla d|)$. $L(d)$ is defined as

$$L(d) = S_{\Delta x}(\phi)\left(1 - \sqrt{(D_x d)^2 + (D_y d)^2}\right), \tag{56}$$

where $S_{\Delta x}(\phi)$ is a smoothed sign function,

$$S_{\Delta x}(\phi) \equiv 2(H_{\Delta x}(\phi) - 1/2), \tag{57}$$

and $D_x d$, $D_y d$ are approximations to $\partial d/\partial x$ and $\partial d/\partial y$, respectively. $D_x d$ is defined as

$$(D_x d)^L_{ij} = D^-_x d_{ij} + \frac{\Delta x}{2} m(D^+_x D^-_x d_{ij}, D^+_x D^-_x d_{i-1,j}) \tag{58}$$

$$(D_x d)^R_{ij} = D^+_x d_{ij} - \frac{\Delta x}{2} m(D^+_x D^-_x d_{ij}, D^+_x D^-_x d_{i+1,j}) \tag{59}$$

$$w^L = (D_x d)^L_{ij} S(\phi) \tag{60}$$

$$w^R = (D_x d)^R_{ij} S(\phi) \tag{61}$$

$$(D_x d)_{ij} = \begin{cases} (D_x d)^L_{ij}, & \text{if } w^L > 0 \text{ and } w^L + w^R > 0, \\ (D_x d)^R_{ij}, & \text{if } w^R < 0 \text{ and } w^L + w^R < 0, \\ 0, & \text{if } w^L < 0 \text{ and } w^R > 0. \end{cases} \tag{62}$$

The function $m(a, b)$ and the difference operators D^-_x and D^+_x, found in (58) and (59), are defined as

$$m(a, b) = \begin{cases} a, & \text{if } |a| \le |b|, \\ b, & \text{otherwise;} \end{cases} \tag{63}$$

$$D^-_x d_{ij} = \frac{d_{ij} - d_{i-1,j}}{\Delta x}, \tag{64}$$

$$D^+_x d_{ij} = \frac{d_{i+1,j} - d_{i,j}}{\Delta x}. \tag{65}$$

Analogous formulas as for $D_x d$ are used to approximate $\partial d/\partial y$ and $\partial d/\partial z$. The discretization described above for $D_x d$ corresponds to a second-order essentially nonoscillatory (ENO) scheme described in detail by [42, 23].

We use an improvement to the redistance step which was developed in [43]. We interpret the term, $S(\phi)$, in (51) as a "constraint" used both to prevent the interface from moving and also to implicitly prescribe boundary conditions at the interface. For discretization purposes we wish to enforce another constraint: the volume filled by each fluid must stay constant when the redistance step is applied. For each grid cell, Ω_{ij}, we define volume as

$$V^k_{ij} = \int_{\Omega_{ij}} H(d^k)\, d\mathbf{x}, \tag{66}$$

where H is the Heaviside function described by (2) and d^k is the value of d at τ^k, the "time" after the kth iteration in the redistance step.

Because volume should not change, we should have $V^k_{ij} = V^0_{ij}$. Nevertheless, if the redistance step slightly changes the location of the zero level set, we then have, for $\tau^k - \tau^0 = O(\Delta x)$,

$$\begin{aligned} V^k_{ij} - V^0_{ij} &\approx (\tau^k - \tau^0) \int_{\Omega_{ij}} \frac{dH_\varepsilon(d^0)}{d\tau}\, d\mathbf{x} \\ &\approx \int_{\Omega_{ij}} H'_\varepsilon(d^0)(d^k - d^0)\, d\mathbf{x}, \end{aligned} \tag{67}$$

where

$$H'_\varepsilon(d) = \begin{cases} 0, & \text{if } |d| > \varepsilon, \\ \frac{1}{2}\left[\frac{1}{\varepsilon} + \frac{1}{\varepsilon}\cos(\pi d/\varepsilon)\right], & \text{if } |d| \le \varepsilon. \end{cases} \tag{68}$$

In order to minimize volume variation, we project the current values of the level set function, denoted as $\widetilde{d_{ij}^k}$, onto new values, denoted as d_{ij}^k, which satisfy

$$\int_{\Omega_{ij}} H'_\varepsilon(d^0)(d^k - d^0)\, d\boldsymbol{x} = 0. \tag{69}$$

If (69) is satisfied then by (67) the volume change will be very small. To implement this projection we assume d_{ij}^k has the form

$$d_{ij}^k = \widetilde{d_{ij}^k} + \lambda_{ij}(\tau^k - \tau^0)H'_\varepsilon(d^0), \tag{70}$$

where λ_{ij} is assumed constant in Ω_{ij}. After substituting (70) into (69), we have

$$\lambda_{ij} = \frac{-\int_{\Omega_{ij}} H'_\varepsilon(d^0)\left(\frac{\widetilde{d^k} - d^0}{\tau^k - \tau^0}\right)dx}{\int_{\Omega_{ij}} \left(H'_\varepsilon(d^0)\right)^2 dx}. \tag{71}$$

Equation (71) is discretized in each cell by using a nine-point stencil to perform the integration. Since λ_{ij} is assumed constant in each cell, the above equation can be solved both explicitly and quickly. The projection step given by (70) is applied after each redistance iteration. In the work of [43] it was shown that the above constraint helps the level set function converge to a distance function while still maintaining the original zero level set.

3.6. *Initialization of the Data*

Specification of the problem must include values for U and ϕ at time $t^0 = 0$ and values for the initial pressure p at time $t^{1/2}$. The pressure is not initially prescribed and must be calculated in an initial iterative step.

To begin the calculation, the initial velocity field is first projected to ensure that it satisfies the divergence constraint at $t = 0$. Then an initial iteration is performed to calculate an approximation to the pressure at $t = \Delta t/2$. If this process were iterated to convergence and the projection were exact, then $U^1 \equiv U^*$ in the first step, because the pressure used in Eq. (16) would in fact be $p^{1/2}$, not $p^{-1/2}$. However, in practice we typically perform only a few iterations, since what is needed for second-order accuracy in Eq. (16) is only a first-order accurate approximation to $p^{n+1/2}$, which in a standard time step is approximated by $p^{n-1/2}$.

In each step of the iteration we follow the procedure described in the above subsections. In the first iteration we use $p^{-1/2} = 0$. At the end of each iteration we have calculated a value of U^1 and a pressure $p^{1/2}$. During the iteration procedure, we discard the value of U^1, but define $p^{-1/2} = p^{1/2}$. Once the iteration is completed, we use the value of $p^{-1/2}$ in Eq. (16) along with the values of U^0 and ϕ^0.

4. ADAPTIVE MESH REFINEMENT

In this section we present the extension of the single-grid algorithm described above to an adaptive hierarchy of nested rectangular grids. The basic adaptive framework is as described by Almgren *et al.* [3].

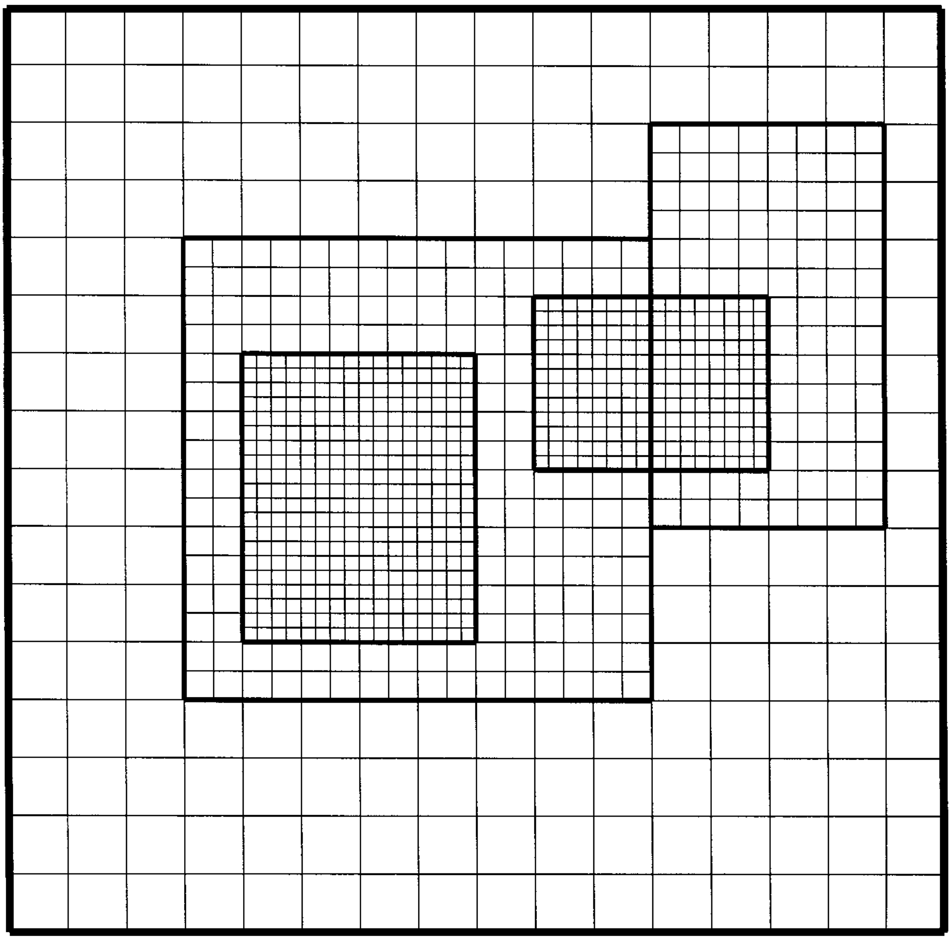

FIG. 2. Diagram of grid structure used in adaptive mesh refinement (AMR). In this example there are three levels. Level 0 has one 16×16 grid. Level 1 has two grids; a 16×16 grid and a 8×14 grid. Level 2 also has two grids; a 16×20 grid and a 16×12 grid. The refinement ratio between levels in this example is 2.

In Fig. 2 we show an example of the grid structure used in adaptive mesh refinement (AMR) [6, 11, 10, 33]. The grid hierarchy is composed of different levels of refinement ranging from coarsest, $\ell = 0$, to finest, $\ell = \ell_{\max}$. The coarsest level, $\ell = 0$, covers the whole computational domain while successively higher levels, $\ell + 1$, lie on top of the level underneath them, level ℓ. Each level is represented as the union of rectangular grid patches of a given resolution. In our computations the refinement ratio between levels is 2. Thus we have $\Delta x^{\ell+1} = \Delta y^{\ell+1} = \Delta z^{\ell+1} = \frac{1}{2}\Delta x^{\ell}$. The grids are properly nested, in the sense that the union of grids at level $\ell + 1$ is contained in the union of grids at level ℓ for $0 \leq \ell < \ell_{\max}$. Furthermore, the containment is strict in the sense that, except at physical boundaries, the level ℓ grids are large enough to guarantee that there is a border at least one level ℓ cell wide surrounding each level $\ell + 1$ grid.

The initial creation of the grid hierarchy and the subsequent regridding operations in which the grids are dynamically changed to reflect changing flow conditions use the same procedures as were used by Bell *et al.* [6] for hyperbolic conservation laws. In the problems we compute here, we shall "tag" cells which contain part of the gas/liquid interface, i.e.

those in which the level set function changes sign. For some problems, in order to generate the finest level of refinement, we require not only that that cells contain the interface, but also that the curvature in those cells exceed a preset threshold in order to be "tagged." Once cells on a specified level are "tagged" for refinement, the grids at the next higher level can be constructed. The tagged cells are grouped into rectangular patches using the clustering algorithm given in Berger and Rigoustsos [12]. These rectangular patches are refined to form the grids at the next level. The process is repeated until either the error tolerance criteria are satisfied or a specified maximum level is reached.

At $t = 0$ the initial data is used to create grids at level 0 through $\ell_{\max}$. Grids have a user-specified maximum size; therefore, more than one grid may be needed to cover the physical domain. As the solution advances in time, the regridding algorithm is called every k steps, k is a user-specified parameter, to redefine grids at levels 1 to $\ell_{\max}$. Level 0 grids remain unchanged throughout the calculation.

When new grids are created at level $\ell + 1$, the data on these new grids are copied from the previous grids at level $\ell + 1$ if possible; otherwise they are interpolated in space from the underlying level ℓ grids. We use conservative interpolation for cell-centered variables $\boldsymbol{U}$ and ϕ [38] and bilinear interpolation for pressure.

We note here that while there is a user-specified limit to the number of levels allowed, at any given time in the calculation there may not be that many levels in the hierarchy; i.e. $\ell_{\max}$ can change dynamically as the calculation proceeds, as long as it does not exceed the user-specified limit.

4.1. *Overview of Time-Stepping Procedure*

An important distinction between the adaptive algorithm presented in this paper and that presented in [3] is the fact that we do not use a "subcycling" timestep procedure.[3] In other words, the data on the coarsest levels is advanced with the same timestep as the data on the finest level. The timestep for each level Δt^{ℓ} is the same as the timestep on the finest level $\Delta t^{\ell_{\max}}$. The procedure to advance $\boldsymbol{U}$ and ϕ on levels 0 thru $\ell_{\max}$ is similar to the Crank–Nicholson procedure presented for the single grid discretization 3.1:

1. *Level set update for* $\phi^{n+1,\ell}$. For levels $0 \le \ell \le \ell_{\max}$,

$$\phi^{n+1,\ell} = \phi^{n,\ell} - \Delta t^{\ell}[\boldsymbol{U} \cdot \nabla\phi]^{n+1/2,\ell}. \tag{72}$$

The evaluation of the nonlinear advection term $[\boldsymbol{U} \cdot \nabla\phi]^{n+1/2,\ell}$ requires that we apply the MAC projection (27) on all levels simultaneously. Details of the composite MAC projection are presented in Section 4.2.

2. *Semi-implicit viscous solve for the intermediate velocity* $\boldsymbol{U}^{*,\ell}$. For levels $0 \le \ell \le \ell_{\max}$, we solve (16). Details of the composite viscous solve are presented in Section 4.3.

3. *Projection step for* $\boldsymbol{U}^{n+1,\ell}$. For levels $0 \le \ell \le \ell_{\max}$ we solve (17). Details of the composite projection step are presented in Section 4.4.

4. *Composite redistance operation on levels* $0 \le \ell \le \ell_{\max}$. Details of the composite redistance step are presented in Section 4.5.

As in the adaptive mesh technique for hyperbolic systems, the extrapolation of $\boldsymbol{U}$ and ϕ to the cell faces at $t^{n+1/2}$, as described by (20) thru (23), can be performed one grid at a

[3] However, we do use a "subcycling" timestep procedure for the composite redistance step; see Section 4.5 for details.

time, with boundary data copied from other fine grids, interpolated from underlying coarse grids, or supplied from physical boundary conditions. The composite MAC projection, composite viscous solve, and composite approximate projection require that the solution be computed on all grids at a level at one time, since these are no longer explicit operations. Boundary data for these solves are interpolated from underlying coarse grids or supplied from physical boundary conditions. The interpolation and solution procedure for these equations are discussed in Sections 4.2, 4.3, and 4.4.

4.2. *Composite MAC Projection*

The composite MAC projection consists of the following steps:

1. Average down unprojected face centered velocities (26) and density, for $\ell = \ell_{\max} - 1 \cdots 0$,

$$\begin{aligned} \boldsymbol{U}^{n+1/2,\ell} &= I^{\ell}_{\ell+1}\boldsymbol{U}^{n+1/2,\ell+1} \\ \rho^{n,\ell} &= I^{\ell}_{\ell+1}\rho^{n,\ell+1}. \end{aligned}$$

The operator $I^{\ell}_{\ell+1}$ represents the averaging down of level $\ell+1$ data onto level ℓ data. The equations for averaging down face-centered velocities and density are

$$u^{\ell}_{i-1/2,j} = \frac{1}{2}\left(u^{\ell+1}_{2i-1/2,2j} + u^{\ell+1}_{2i-1/2,2j+1}\right) \tag{73}$$

$$v^{\ell}_{i,j-1/2} = \frac{1}{2}\left(v^{\ell+1}_{2i,2j-1/2} + v^{\ell+1}_{2i+1,2j-1/2}\right) \tag{74}$$

$$\rho^{\ell}_{i,j} = \frac{1}{4}\left(\rho^{\ell+1}_{2i,2j} + \rho^{\ell+1}_{2i+1,2j} + \rho^{\ell+1}_{2i,2j+1} + \rho^{\ell+1}_{2i+1,2j+1}\right). \tag{75}$$

Analogous equations are used for three-dimensional problems.

2. Perform a single level MAC-project on each level for $\ell = 0 \cdots \ell_{\max}$

$$D^{\mathrm{MAC}}\left(\frac{1}{\rho^{n,\ell}}G^{\mathrm{MAC}}p^{\mathrm{MAC},\ell}\right) = D^{\mathrm{MAC}}(\boldsymbol{U}^{n+1/2,\ell})$$

$$\begin{aligned} u^{\mathrm{ADV},\ell}_{i+1/2,j} &= u^{n+1/2,\ell}_{i+1/2,j} - \frac{1}{\rho^{n,\ell}_{i+1/2,j}}\left(G^{\mathrm{MAC}}_x p^{\mathrm{MAC},\ell}\right)_{i+1/2,j} \\ v^{\mathrm{ADV},\ell}_{i,j+1/2} &= v^{n+1/2,\ell}_{i,j+1/2} - \frac{1}{\rho^{n,\ell}_{i,j+1/2}}\left(G^{\mathrm{MAC}}_y p^{\mathrm{MAC},\ell}\right)_{i,j+1/2}. \end{aligned}$$

Remarks. • The solver for a single level MAC-project is the same as that described for a single grid MAC-project (27) except that on each level, we have to solve over a collection of grids. Also, Dirichlet boundary conditions for pressure have to be specified at boundaries between levels ℓ and $\ell - 1$.

• Minion *et al.* have presented a multigrid-based composite MAC projection algorithm in [33] for constant density problems. They solve over all levels simultaneously. For our problems, in which the density ratio is 816:1, we have had convergence problems using only multigrid for (43); thus we resort to solving a level at a time using the multigrid preconditioned conjugate gradient method and then synchronizing levels $\ell = 0 \cdots \ell_{\max}$ as described in the next step.

3. Synchronize newly MAC-projected edge velocities to be discretely divergence-free across coarse-fine grid boundaries for $\ell = \ell_{\max} - 1 \cdots 0$

$$\begin{aligned}
\boldsymbol{U}^{\mathrm{ADV},\ell} &= I_{\ell+1}^{\ell} \boldsymbol{U}^{\mathrm{ADV},\ell+1} \\
D^{\mathrm{MAC}}\left(\frac{1}{\rho^{n,\ell}} G^{\mathrm{MAC}} \tilde{p}^{\mathrm{MAC},\ell}\right) &= D^{\mathrm{MAC}}(\boldsymbol{U}^{\mathrm{ADV},\ell}) \quad (\text{“MAC-synchronization”}) \\
\tilde{u}_{i+1/2,j}^{\mathrm{ADV},\ell} &= u_{i+1/2,j}^{\mathrm{ADV},\ell} - \frac{1}{\rho_{i+1/2,j}^{n,\ell}} \left(G_x^{\mathrm{MAC}} \tilde{p}^{\mathrm{MAC},\ell}\right)_{i+1/2,j} \\
\tilde{v}_{i,j+1/2}^{\mathrm{ADV},\ell} &= v_{i,j+1/2}^{\mathrm{ADV},\ell} - \frac{1}{\rho_{i,j+1/2}^{n,\ell}} \left(G_y^{\mathrm{MAC}} \tilde{p}^{\mathrm{MAC},\ell}\right)_{i,j+1/2} \\
u_{i+1/2,j}^{\mathrm{correct},\ell} &= \tilde{u}_{i+1/2,j}^{\mathrm{ADV},\ell} - u_{i+1/2,j}^{\mathrm{ADV},\ell} \\
v_{i,j+1/2}^{\mathrm{correct},\ell} &= \tilde{v}_{i,j+1/2}^{\mathrm{ADV},\ell} - v_{i,j+1/2}^{\mathrm{ADV},\ell};
\end{aligned}$$

for $\ell' = \ell + 1 \cdots \ell_{\max}$

$$\tilde{\boldsymbol{U}}^{\mathrm{ADV},\ell'} = \tilde{\boldsymbol{U}}^{\mathrm{ADV},\ell'} + I_{\ell}^{\ell'} \boldsymbol{U}^{\mathrm{correct},\ell}.$$

Remarks. • Homogeneous Dirichlet boundary conditions for $\tilde{p}^{\mathrm{MAC},\ell}$ are enforced at boundaries between levels ℓ and $\ell - 1$.

• The only contribution to the right hand side of the “MAC-synchronization” step will be at cells on the immediate coarse grid side of the boundaries separating levels $\ell + 1$ and ℓ. This is because the averaging down procedure for the face-centered velocities (73, 74) preserves the discrete divergence free property; i.e., if $D^{\mathrm{MAC}} \boldsymbol{U}^{\mathrm{ADV},\ell+1} = 0$ then we also have $D^{\mathrm{MAC}} \boldsymbol{U}^{\mathrm{ADV},\ell} = 0$.

• The interpolation operator $I_{\ell}^{\ell'}$ also preserves the discrete divergence free property. The equations for interpolation are described for the case when $\ell' = \ell + 1$ as

$$u_{2i-1/2,2j}^{\ell+1} = u_{i-1/2,j}^{\ell} \tag{76}$$

$$u_{2i-1/2,2j+1}^{\ell+1} = u_{i-1/2,j}^{\ell} \tag{77}$$

$$u_{2i+1/2,2j}^{\ell+1} = \frac{1}{2}\left(u_{i-1/2,j}^{\ell} + u_{i+1/2,j}^{\ell}\right) \tag{78}$$

$$u_{2i+1/2,2j+1}^{\ell+1} = \frac{1}{2}\left(u_{i-1/2,j}^{\ell} + u_{i+1/2,j}^{\ell}\right). \tag{79}$$

Analogous equations are used for the other velocity components and also for three-dimensional problems.

• If the resulting corrected fine grid velocities $\tilde{\boldsymbol{U}}^{\mathrm{ADV},\ell+1}$ are averaged down onto level ℓ, there will be no change in $\tilde{\boldsymbol{U}}^{\mathrm{ADV},\ell}$. Furthermore, $D^{\mathrm{MAC}} \tilde{\boldsymbol{U}}^{\mathrm{ADV},\ell} = 0$. This is because our averaging operator $I_{\ell+1}^{\ell}$ and our interpolation operator $I_{\ell}^{\ell'}$ preserve the discrete divergence.

• In all of our computations the ratio of $\|\boldsymbol{U}^{\mathrm{correct},\ell}\|_2$ to $\|\boldsymbol{U}^{\mathrm{ADV},\ell}\|_2$ is less than 10^{-8}. This is because we have averaged down the unprojected MAC velocities prior to our composite solve. This is only possible since we do not use a “subcycling” time step procedure (see Section 4.5 for outline of the “subcycling” time step procedure).

• In our computations shown in Fig. 8, the synchronization procedure took 5% of the total CPU time and the full composite MAC project procedure, including the synchronization procedure, took 32% of the total CPU time. The synchronization procedure took a comparable amount of time for all of our other computations too.

4.3. *Composite Semi-Implicit Solve*

In this section we describe how we solve (16) over multiple levels. The composite semi-implicit solve consists of the following steps:

1. Average down the time centered density for $\ell = \ell_{\max} - 1 \cdots 0$

$$\rho^{n+1/2,\ell} = I_{\ell+1}^{\ell} \rho^{n+1/2,\ell+1}.$$

2. On each level, form the right-hand side of (16), excluding the viscous terms, for $\ell = 0 \cdots \ell_{\max}$

$$\boldsymbol{V}^{n+1/2,\ell} = \boldsymbol{U}^{n,\ell} + \Delta t\left(-[(\boldsymbol{U}\cdot\nabla)\boldsymbol{U}]^{n+1/2,\ell} - \frac{Gp^{n-1/2,\ell}}{\rho^{n+1/2,\ell}} - \frac{\mathcal{M}^{n+1/2,\ell}}{\rho^{n+1/2,\ell}} + \boldsymbol{F}\right).$$

3. Average down $\boldsymbol{V}^{n+1/2,\ell}$ for $\ell = \ell_{\max} - 1 \cdots 0$

$$\boldsymbol{V}^{n+1/2,\ell} = I_{\ell+1}^{\ell} \boldsymbol{V}^{n+1/2,\ell+1}.$$

4. Perform a single level semi-implicit viscous solve (16) on each level, solving for $\boldsymbol{U}^{*,\ell}$ for $\ell = 0 \cdots \ell_{\max}$

$$\boldsymbol{U}^{*,\ell} - \Delta t\frac{\mathcal{L}^{*,\ell} + \mathcal{L}^{n,\ell}}{2\rho^{n+1/2,\ell}} = \boldsymbol{V}^{n+1/2,\ell} \quad (\text{“Single-level viscous solve”}).$$

5. Average down $\boldsymbol{U}^{*,\ell}$ for $\ell = \ell_{\max} - 1 \cdots 0$

$$\boldsymbol{U}^{*,\ell} = I_{\ell+1}^{\ell} \boldsymbol{U}^{*,\ell+1}.$$

The issues in solving the “single level viscous solve” as opposed to a single grid viscous solve (16) are described in detail in Section 3.5 of [3]. The boundary conditions for $\boldsymbol{U}^{*,\ell}$ are homogeneous Dirichlet at a solid wall and homogeneous Neumann at outflow boundaries. At a coarse–fine grid boundary, i.e. the boundary separating levels ℓ and $\ell - 1$, the solution is specified by quadratic interpolation from the coarser level. Here we address the only additional issue, that of how to provide boundary conditions for a nine-point stencil in two dimensions rather than the five-point stencil. The extension to three dimensions follows analogously. The difficulty here is in how to define the value at each corner point of the stencil when that point lies outside the fine grid.

There are three possibilities for the corner ghost cell: (1) it lies in another fine grid, in which case the value is supplied by the other fine grid; (2) it lies outside the physical boundary, in which case the value is supplied by the physical boundary conditions; or (3) the value must be interpolated from the coarse grids. The interpolation scheme for such ghost cells is described in detail in [3] when the ghost cell is aligned with a row of fine grid cells, such as is always the case with a five-point stencil and is the case for the nine-point stencil, except at the corner of the grid. At the fine grid corners, rather than interpolate between fine grid and coarse grid points along a diagonal, the ghost cell value is filled by extrapolating from ghost cell values along one of the edges intersecting that corner.

4.4. *Composite Approximate Projection*

In this section we describe how we compute the approximate projection (17) over multiple levels.

Since $\boldsymbol{U}^{*,\ell}$ and $\rho^{n+1/2,\ell}$ were averaged down in the "Composite Semi-Implicit Solve" (Section 4.3), we do not need to average down these quantities prior to the composite approximate projection. The composite approximate projection consists of the following steps:

1. Perform a single level approximate projection step (17) on each level, solving for $\boldsymbol{U}^{n+1,\ell}$ for $\ell = 0 \cdots \ell_{\max}$

$$\frac{\boldsymbol{U}^{n+1,\ell} - \boldsymbol{U}^{n,\ell}}{\Delta t} = \mathcal{P}\left(\frac{\boldsymbol{U}^{*,\ell} - \boldsymbol{U}^{n,\ell}}{\Delta t}\right) \quad \text{("single-level approximate projection").}$$

2. Average down $\boldsymbol{U}^{n+1,\ell}$ for $\ell = \ell_{\max} - 1 \cdots 0$

$$\boldsymbol{U}^{n+1,\ell} = I_{\ell+1}^{\ell} \boldsymbol{U}^{n+1,\ell+1}.$$

Remarks. • The "single level approximate projection" is the same as the single grid approximate projection described in Section 3.4.1, except that on each level we must solve over a collection of grids. Also, Dirichlet boundary conditions for pressure have to be specified at boundaries between level ℓ and level $\ell - 1$.

• Projecting on each level individually is contrary to the elliptic nature of the governing Eqs. (3) and (4). That is to say, the solution on coarse grids depends not only on local conditions, such as would be the case if one were solving purely hyperbolic equations, but also on nonlocal conditions, such as data lying underneath fine grids. In our implementation, we advance all levels with the same timestep; this allows us to average down $(\boldsymbol{U}^{*,\ell} - \boldsymbol{U}^{n,\ell})/\Delta t$ prior to solving the projection equation. By averaging down prior to the projection equation, we minimize the decoupling error generated by solving one level at a time. The ratio of $\|\boldsymbol{U}^{\text{correct},\ell}\|_2$ to $\|\boldsymbol{U}^{\ell}\|_2$ (see Section 4.2 for the definition of $\|\boldsymbol{U}^{\text{correct},\ell}\|_2$) was less than 10^{-8} for all our computations. In Sections 5.1.1 and 5.3, we did computations on both an adaptive grid and a single uniformly fine grid; the adaptive results showed very good agreement with the single grid results.

4.5. *Composite Redistance Step*

In this section, we describe how to solve (51) on levels $0 \le \ell \le \ell_{\max}$. If ε is the thickness of our interface, then we solve (51) for $\tau = 0$ to $\tau = \varepsilon$.

The procedure to advance d is a subcycling procedure; the fictitious time step on each level satisfies $\Delta\tau^{\ell} = \Delta x^{\ell}/2$. We outline the procedure to advance the solution of (51) on level ℓ from "time" τ to time $\tau + \Delta\tau^{\ell}$. This procedure is recursive with respect to refinement level. Further details on subcycling are given in [3, 38, 10].

ALGORITHM (Advance solution on level ℓ from "time" τ to $\tau + \Delta\tau^\ell$).

1. Advance solution on each level ℓ grid from 'time' τ to $\tau + \Delta\tau^\ell$. Boundary conditions for d are supplied from level $\ell - 1$ if $\ell > 0$, and from the physical domain boundaries.
2. Advance the solution on level $\ell + 1$ from τ to $\tau + \Delta\tau^\ell/2$, then from $\tau + \Delta\tau^\ell/2$ to $\tau + 2\Delta\tau^\ell/2$.
3. Average the solution from level $\ell + 1$ onto level ℓ.

We remark that the above "subcycling" procedure is a recursive procedure because in order to advance the solution on level $\ell + 1$ above, one must recursively call the above algorithm again with ℓ replaced by $\ell + 1$, and then again with $\ell + 1$ replaced by $\ell + 2 \ldots$.

5. NUMERICAL EXAMPLES

We present air/water computations on a two-dimensional axisymmetric (r-z) grid and on a fully three-dimensional grid. We validate our method via convergence checks, direct comparison with other numerical methods, and comparison with experiments.

5.1. *Convergence Checks*

We measure the order of accuracy of our method by comparing the relative error between successively refined computations. The relative error is defined as

$$E(t) = \sum_{i,j} \int_{\Omega_{ij}} |H(\phi_c(t)) - H(\phi_f(t))|\, dx, \tag{80}$$

where ϕ_c is the level set function from a coarser computation and ϕ_f is the level set function from the refined computation. The integral in (80) is approximated by partitioning Ω_{ij} into 100×100 rectangles and then applying the midpoint rule. The values of ϕ_c and ϕ_f at the midpoint of each rectangle are obtained via bilinear interpolation.

We also measure the volume to check that volume conservation is attained as the grid is refined. The volume is measured as

$$V(t) = \sum_{i,j} \int_{\Omega_{ij}} H(\phi(t))\, dx. \tag{81}$$

5.1.1. Inviscid gas bubble I. We compute the evolution of an axisymmetric rising inviscid air bubble in water with surface tension. Inviscid gas bubbles have been studied computationally by [40, 31, 13, 47]. As a remark, [40] also used AMR in their computations of a shock-bubble interaction. In our first test problem, the density ratio is 816:1 and the Weber number is 200. In Fig. 3, we display the bubble at times $t = 0.0$, $t = 1.2$, and $t = 1.3$. The spatial mesh size on the finest level is $\Delta x^{\ell_{\max}} = 6/512$ and the interfacial thickness parameter is $\varepsilon = 3\Delta x^{\ell_{\max}}$. The solid line represents results for the same problem using the boundary integral method [44] with 240 points placed on the free surface. In order to compare with the boundary integral method, we use far-field boundary conditions on all sides of the domain except at $r = 0$; i.e., we assume that the pressure on the walls is $p = \rho_l z/Fr$.

In Table I, we show that the solution converges at a rate of $O(h^{1.5})$ as progressively finer levels are added. We also compared the adaptive results in which $\Delta x^{\ell_{\max}} = 6/256$ to the

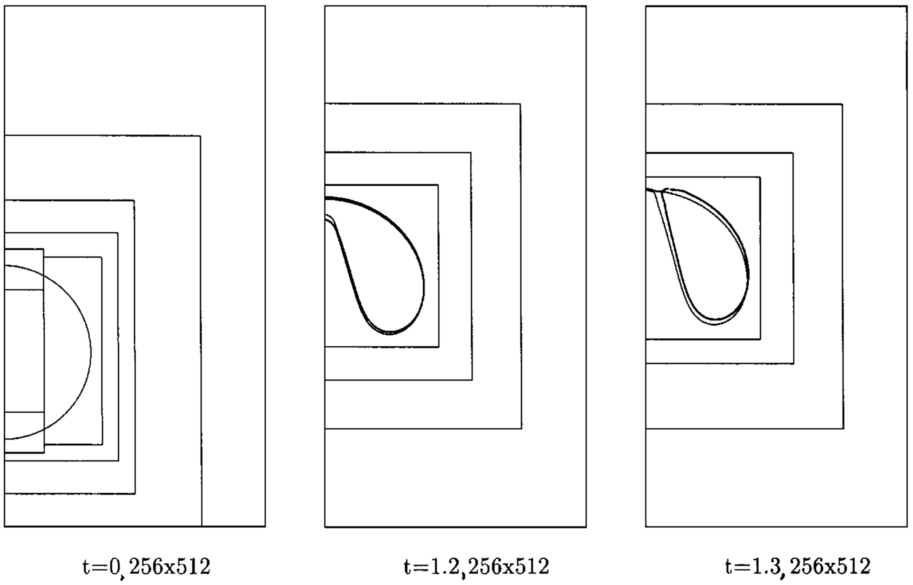

FIG. 3. Spherical gas bubble in liquid: density ratio 816:1, We = 200. Results computed using the adaptive levelset method (thin contour) are compared to results computed using the boundary integral method (thick contour).

single grid results in which $\Delta x = 6/256$ over the whole domain. The difference between these two computations at $t = 1.3$ is 0.0036 which is considerably less than the errors listed in Table I. The speedup of the adaptive computation over the corresponding single grid computation was 2.5.

In order to study the behavior of a rising gas bubble during a change in topology, we added an extra level of adaptivity only at regions of high curvature. The results for this computation are compared to the results without the extra level of adaptivity in Fig. 4. As depicted by Fig. 4, the bubble never pinches off. In fact, it appears that if we refined further and further, the bubble would never break; but the bottom surface would get arbitrarily close to the top surface. We remark that the boundary integral method seems to indicate that the bubble should pinch off in finite time. But, the boundary integral method assumes zero density in the air. In the work of Best [13], they also show pinch off in finite time using the boundary integral method, but they retain an infinitely thin strip at the top of the resulting toroidal bubble. This thin strip is very similar to the connecting line that we show in Fig. 4.

TABLE I
Convergence Study for W = 200 and $\alpha = 3$ at $t = 1.3$

$\Delta x^{\ell\max}$	$V(1.3)$	$E(1.3)$	Order
6/64	4.09	N/A	N/A
6/128	4.14	0.184	N/A
6/256	4.16	0.065	1.5
6/512	4.19	0.023	1.5

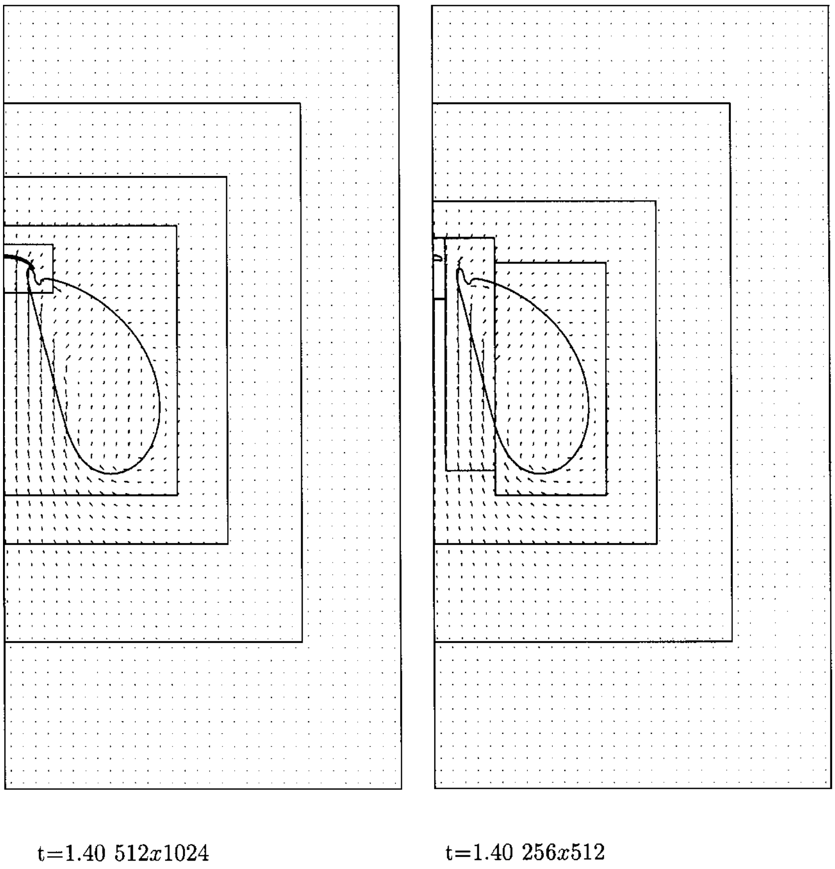

FIG. 4. Spherical, inviscid gas bubble in liquid; density ratio 816:1, We = 200. For results on the left, an extra level of adaptivity is added at region of high high curvature.

5.1.2. Inviscid gas bubble II. In this section, we compute the same problem as in Section 5.1.1, except that we set the Weber number to 10 instead of 200. In Fig. 5, we display the bubble at times $t = 1.7$ and $t = 2.0$. The spatial mesh size on the finest level is $\Delta x^{\ell\max} = 6/512$ and the interfacial thickness parameter is $\varepsilon = 3\Delta x^{\ell\max}$. The solid line represents results for the same problem using the boundary integral method [44] with 240 points placed on the free surface. As in Section 5.1.1, we use far-field boundary conditions on all sides of the domain except at $r = 0$. In Tables II and III, we show convergence results at $t = 1.7$ and $t = 2.0$ when progressively finer levels are added.

TABLE II
Convergence Study for W = 10 and α = 3 at t = 1.7

$\Delta x^{\ell\max}$	$V(1.7)$	$E(1.7)$	Order
6/64	4.05	N/A	N/A
6/128	4.12	0.233	N/A
6/256	4.15	0.083	1.5
6/512	4.17	0.029	1.5

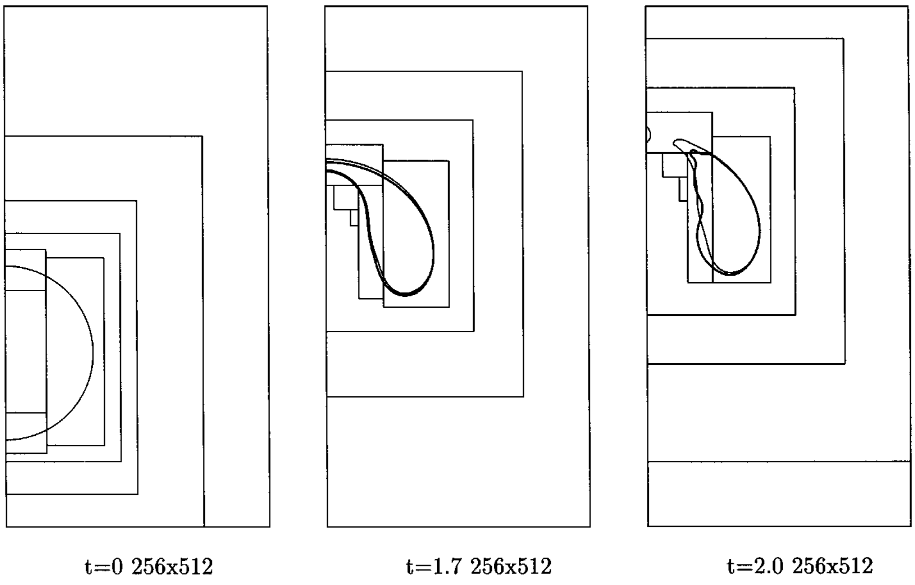

FIG. 5. Spherical gas bubble in liquid: density ratio 816:1, We = 10. Results computed using the adaptive levelset method (thin contour) are compared to results computed using the boundary integral method (thick contour).

As in Section 5.1.1, we add an extra level of adaptivity only at regions of high curvature. The results for this computation are shown in Fig. 6. By contrast with the results for $W = 200$ in Section 5.1.1, it appears from the refined results here for $W = 10$ that the bubble will pinch off no matter how much we refine the grid. In Table IV, we show the pinch-off time for successively refined computations of the $W = 10$ case.

5.1.3. Zero gravity drop oscillation. In this section we compute axisymmetric zero-gravity drop dynamics and compare these with the linearized drop oscillation solutions of Lamb [29, Section 275]. The interfacial position of the drop is shown to be

$$x(\theta, t) = a + \epsilon P_n(\cos(\theta)) \sin(\omega_n t),$$

where

$$\omega_n^2 = \frac{1}{W} \frac{n(n-1)(n+1)(n+2)}{a^3(n+1+n\rho_2/\rho_1)}$$

TABLE III
Convergence Study for W = 10 and α = 3 at t = 2.0

$\Delta x^{\ell\max}$	$V(2.0)$	$E(2.0)$	Order
6/64	4.06	N/A	N/A
6/128	4.12	0.422	N/A
6/256	4.14	0.166	1.3
6/512	4.16	0.072	1.2

TABLE IV
Convergence of Pinch-Off Time for W = 200 and $\alpha = 3$

$\Delta x^{\ell_{\max}}$	Pinch-off time
6/64	2.439
6/128	2.007
6/256	1.912
6/512	1.919

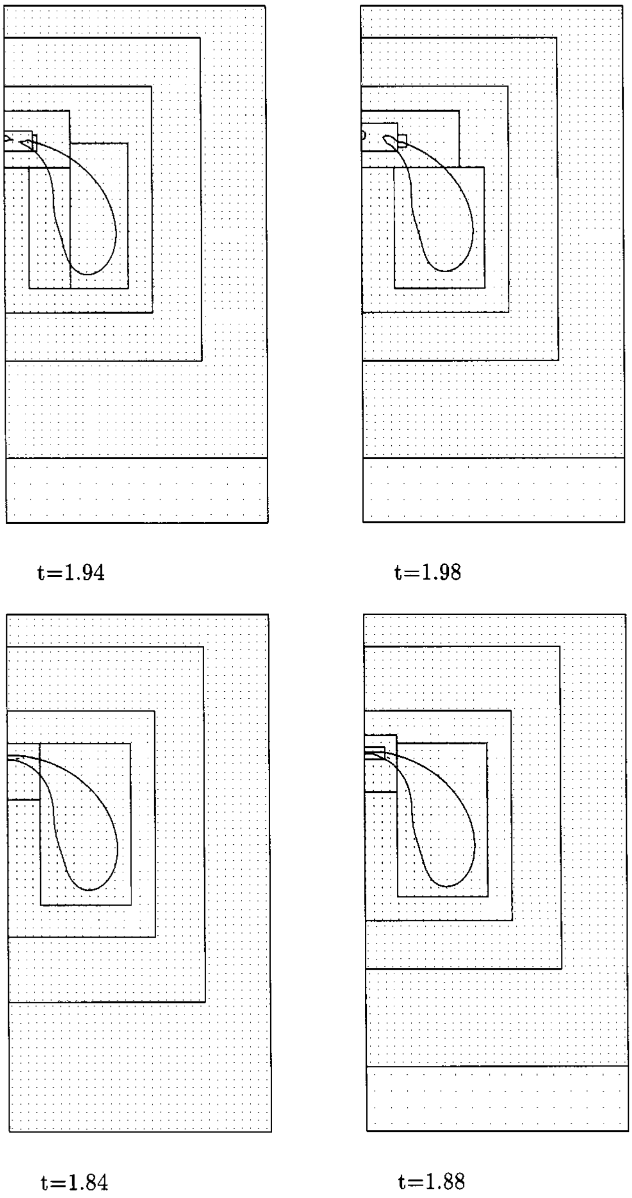

FIG. 6. Spherical inviscid gas bubble in liquid; density ratio 816:1, We = 10. Effective fine grid resolution 512×1024.

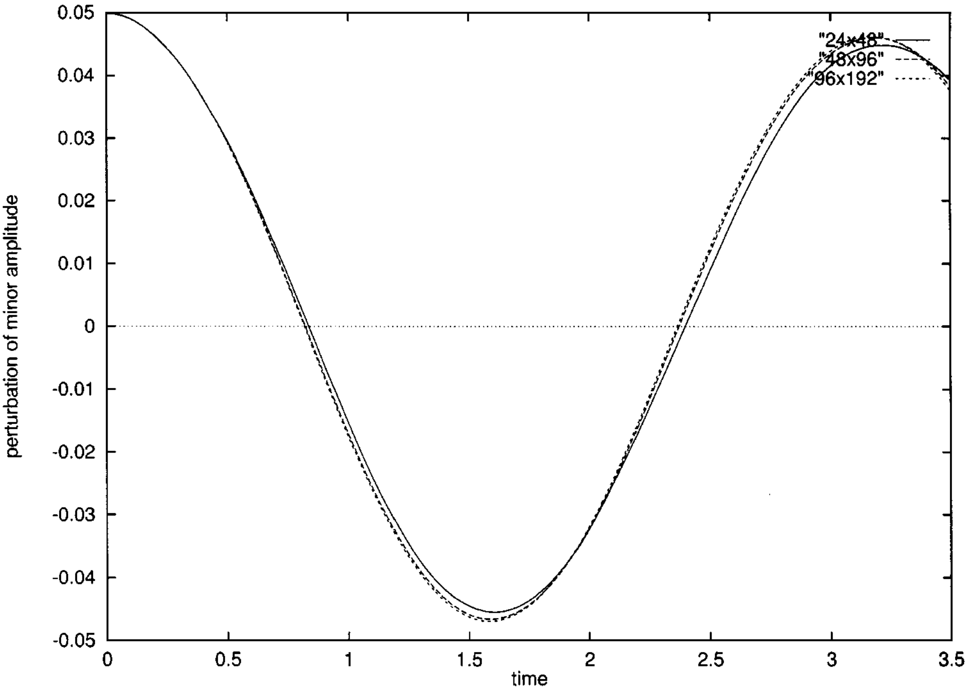

FIG. 7. Perturbation in minor amplitude for zero gravity drop oscillations. W = 2, R = 200, density ratio 100:1, viscosity ratio 100:1.

and P_n is the Legendre polynomial of order n. If viscosity is present, the amplitude is proportional to $e^{-t/\tau}$, where

$$\tau = \frac{a^2 R}{(2n+1)} \frac{n + (\rho_2/\rho_1)(n+1)}{n(n-1) + (n+1)(n+2)\mu_2/\mu_1}.$$

This equation is derived following the approach outlined in Lamb [29, Section 355].

We compute the evolution of a drop with $a = 1$, $\mu_2/\mu_1 = 0.01$, W = 2, $\rho_2/\rho_1 = 0.01$, and R = 200. The initial free surface is given by $x(\theta, 0)$, with $\epsilon = 0.05$ and $n = 2$. With these parameters we find $\omega_2 = 2.00$ and $\tau = 38.3$. The fluid domain is $\Omega = \{(x, y) \mid 0 \le x \le 2$ and $0 \le y \le 4\}$ and the coarse grid size is 24×48. We compare results as succeeding levels of adaptivity are added. The interfacial thickness parameter α is $2\Delta x^{\ell_{\max}}$. The results of our computations are shown in Fig. 7, where we display the perturbation in the minor axis for varying levels of resolution. The average dimensionless period is 3.17 and the expected linearized period is π. The L^1 error between succeeding resolutions is measured as

$$\int_0^{\pi} |x_h(0, t) - x_{2h}(0, t)|\, dt$$

and the L^∞ error is measured as

$$\max_{0 \le t \le \pi} |x_h(0, t) - x_{2h}(0, t)|.$$

We display our computed errors in Table V.

TABLE V
Convergence Study Zero Gravity Drop Oscillations W = 2, Re = 200, Minor Axis

$\Delta x^{\ell \max}$	L_1	L_∞	Period	Order (L_∞)
4/48	N/A	N/A	3.22	N/A
4/96	1.25E-3	3.04E-3	3.18	N/A
4/192	3.50E-4	6.89E-4	3.17	2.1

5.2. *Comparison with Experiments*

We compute the evolution of an axisymmetric rising gas bubble in a viscous liquid. The density ratio is 714:1 and the viscosity ratio is 6667:1. The Reynolds number, Froude number and Weber number are 9.7, 0.78, and 7.6, respectively. These are the same parameters used in bubble experiments by Hnat and Buckmaster [25] and used in steady bubble computations by Ryskin and Leal [41]. In Fig. 8, we show the free surface of the rising bubble. For this problem, we have an extra level of adaptivity in the region of highest curvature. In Fig. 9, we compare the volume of the bubble when computed with the extra adaptivity as opposed to without. In Fig. 10, we display the position of the center of mass of the bubble

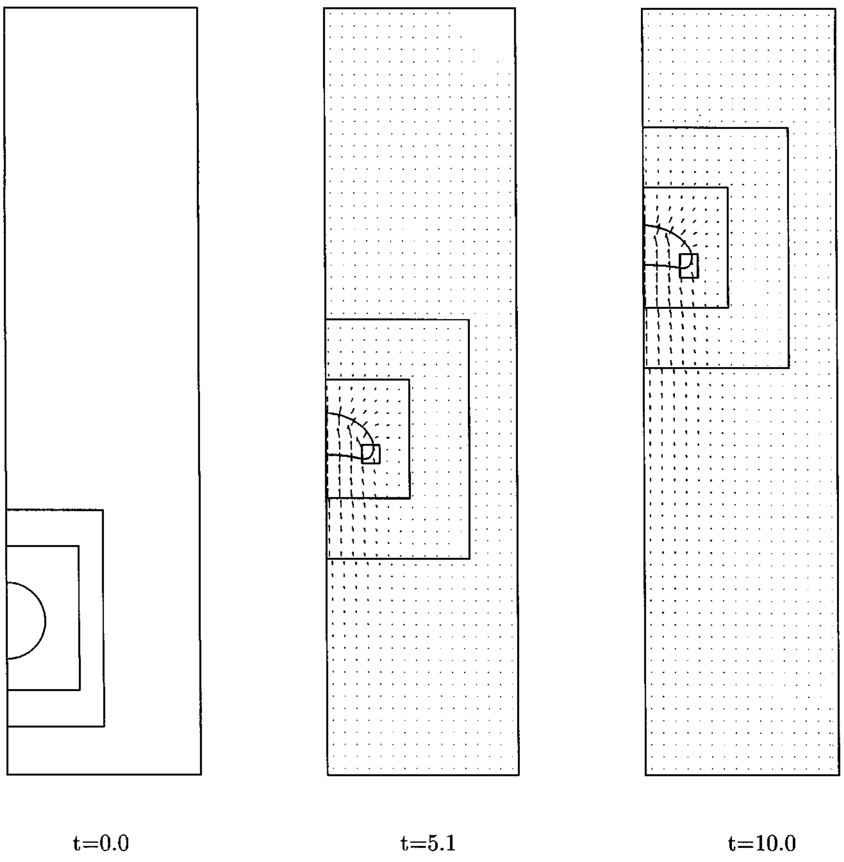

FIG. 8. Rise of an initially spherical gas bubble in viscous liquid. An extra level of adaptivity is automatically added when corner forms in the ensuing cap bubble; density ratio 714:1, viscosity ratio 6667:1, Re = 9.7, We = 7.6, Fr = 0.78.

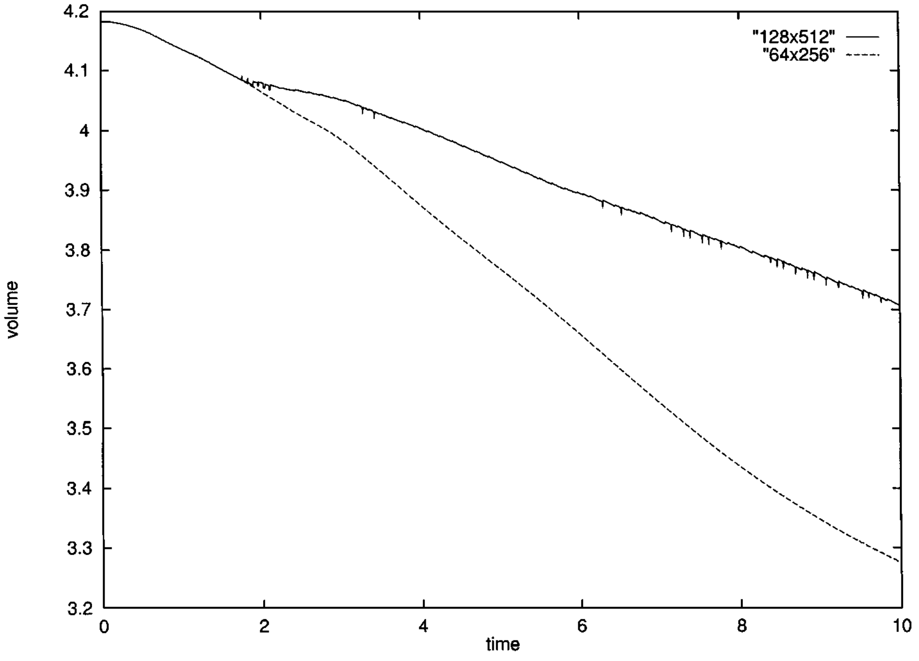

FIG. 9. Plot of mass of a rising cap bubble vs time. Data corresponding to "128 × 512" was computed in which an extra level of adaptivity was added when the corner formed on the cap (about $t = 2$). For data corresponding to "64 × 256," an extra level of adaptivity was not added.

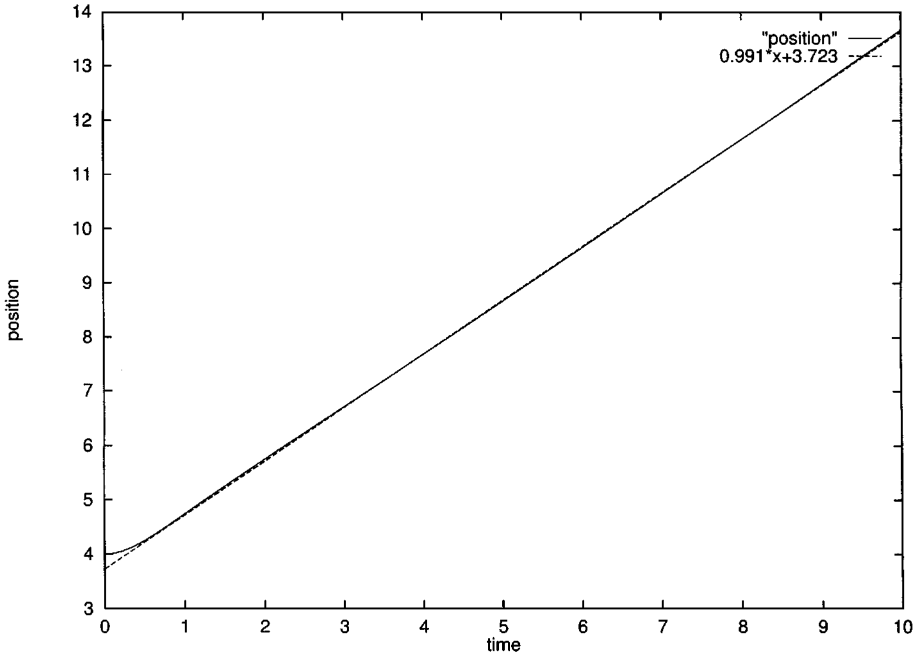

FIG. 10. Plot of the center of mass of a rising cap bubble vs time. We compare this data with the linear best fit for $2 < \text{time} < 10$. Expected slope is 1.

versus time. The average dimensionless rise speed for this case was 0.99 which differs from the experiments by 1%. We believe that some of the error in the computed steady rise-speed is attributable to the fact that we compute in a limited domain and use far-field boundary conditions. When our computation was run in a domain one quarter the size (each dimension cut in half) the average dimensionless rise speed was 1.03, a difference of 3% from the experiments. The advantage of adaptivity here is that enlarging the domain adds cells only at the coarsest level; the fine grids covering the bubble remain the same size. The time to run from $t = 0$ to 6.25 in the large domain, 5×20, was only 30% slower than the cpu time used for the small domain, 2.5×10.

As a remark, our method has the capability of adapting *only* the level set function for additional levels $\ell_{\max} < \ell \leq \ell_{\max}^{\max}$. The level set equation (72) is solved on the additional levels using the subcycling procedure described in Section 4.5. In order to solve (72), we interpolate the face-based advection velocities (28) and cell-centered velocities in space and time using bilinear interpolation. The composite redistance step remains unchanged, except that it covers levels $0 \leq \ell \leq \ell_{\max}^{\max}$. In Fig. 11, we show the free surface for a rising viscous gas bubble; four levels of adaptivity are used in which only the level set function is adapted at the last two levels. The last two levels of adaptivity are located at regions of

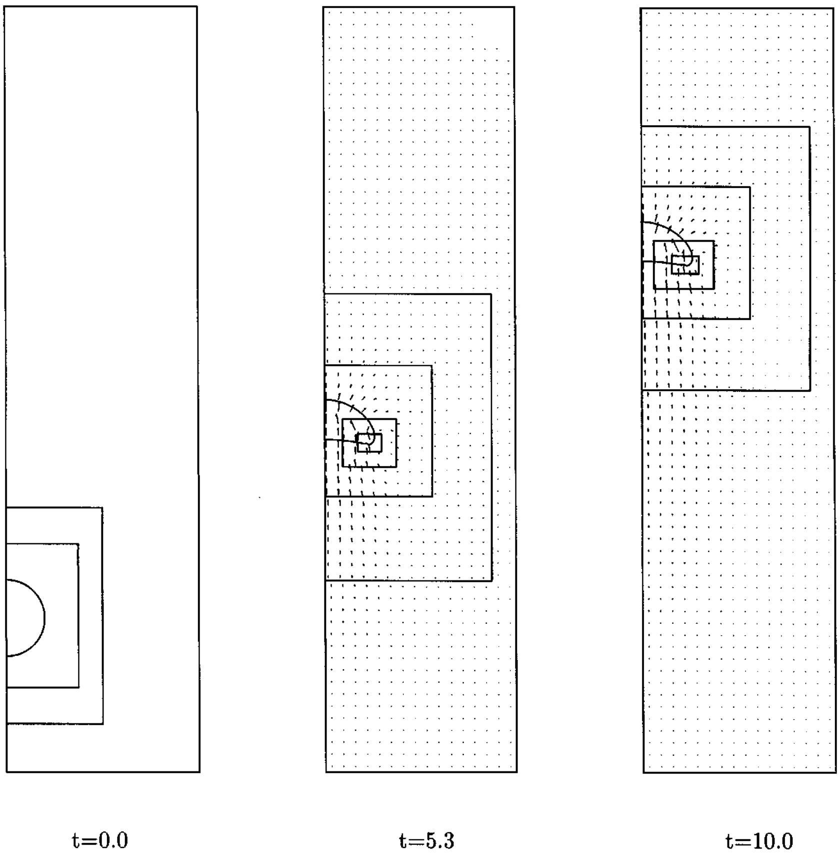

FIG. 11. Rise of an initially spherical gas bubble in viscous liquid. Two extra levels of adaptivity for only the level set function are automatically added when corner forms in the ensuing cap bubble; density ratio 714:1, viscosity ratio 6667:1, Re = 9.7, We = 7.6, Fr = 0.78.

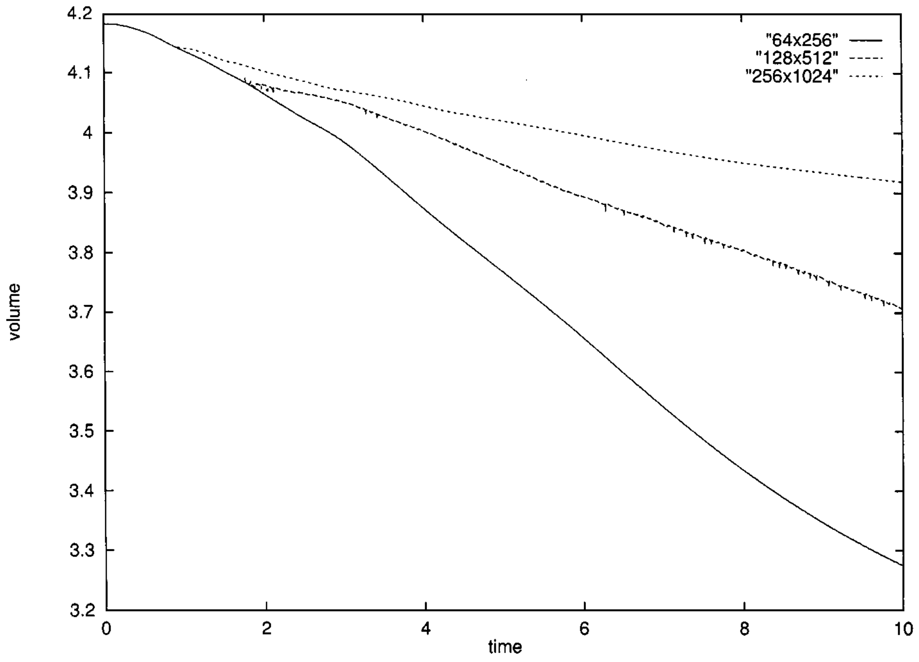

FIG. 12. Plot of mass of a rising cap bubble vs time. Data corresponding to "256 × 1024" was computed in which two extra levels of adaptivity for only the level set function were added when the corner formed on the cap.

high curvature. In Fig. 12, we compare the volume for the computation shown in Fig. 11 to the results shown in Fig. 9. There is a 7% increase in computation time due to computing the level set advection equation and level set redistance operation on the last two levels of adaptivity.

5.3. *Impact of Drop on Water Surface I*

We compute the impact of an axisymmetric water drop on a pool of water along with the "splash" that comes afterward. With the level set method, we automatically handle the merge of the drop with the pool of water and also the breakup of the water splash. In our computations, we use dimensionless parameters based on the impact velocity U and the radius of the drop L. In Fig. 17, we show results using $L = 1$ mm and $U = 4.0$ m/s. In Fig. 18, we show results using $L = 1$ mm and $U = 7.6$ m/s. The dimensionless impact velocity is 1; we accelerate the drop with a fictitious gravitational force term $1/\mathrm{Fr} = 1/2$ for a total dimensionless time 2. At dimensionless time $t = 2$, the drop will be traveling with dimensionless speed of 1 and will have traveled a dimensionless distance of 1 (which is the initial distance between the drop and the pool). For $U = 4.0$ m/s we have $\mathrm{Re} = 3518$, $\mathrm{Fr} = 1633$, and $\mathrm{We} = 220$. For $U = 7.6$ m/s we have $\mathrm{Re} = 6684$, $\mathrm{Fr} = 5895$, and $\mathrm{We} = 794$. As suggested by the difference in Weber number, the spray in the results for $U = 4.0$ m/s (Fig. 17) coagulates more at the tip than the results for $U = 7.6$ m/s (Fig. 18).

We recomputed the drop impact problem on a single uniform fine 128×256 grid for the case when $U = 7.6$ m/s (Fig. 19). In Fig. 20, we compare the uniform fine grid results to the adaptive results. Although the internal memory savings for using an adaptive grid was 4:1, the speedup was only 1.7. We believe that this is due to the elliptic solvers for the approximate projection step and the MAC projection step. The elliptic solvers solve over a collection of grids at each level. The multigrid preconditioner used by the elliptic solvers can only coarsen as much as determined by the smallest grid. On the bottom multigrid level, we must resort to a Gauss–Seidel preconditioned conjugate gradient method. Alternative elliptic solvers [33, 26] solve on all levels simultaneously, thus avoiding the limitation set for the bottom multigrid level. Unfortunately, these solvers are based on a straight multigrid method, in which we have experienced convergence problems for air/water flows with changes in topology.

5.4. *Impact of Drop on Water Surface II*

We compute the impact and subsequent bubble entrainment of an axisymmetric water drop on a pool of water. In the work by Oguz and Prosperetti [36], the boundary integral method

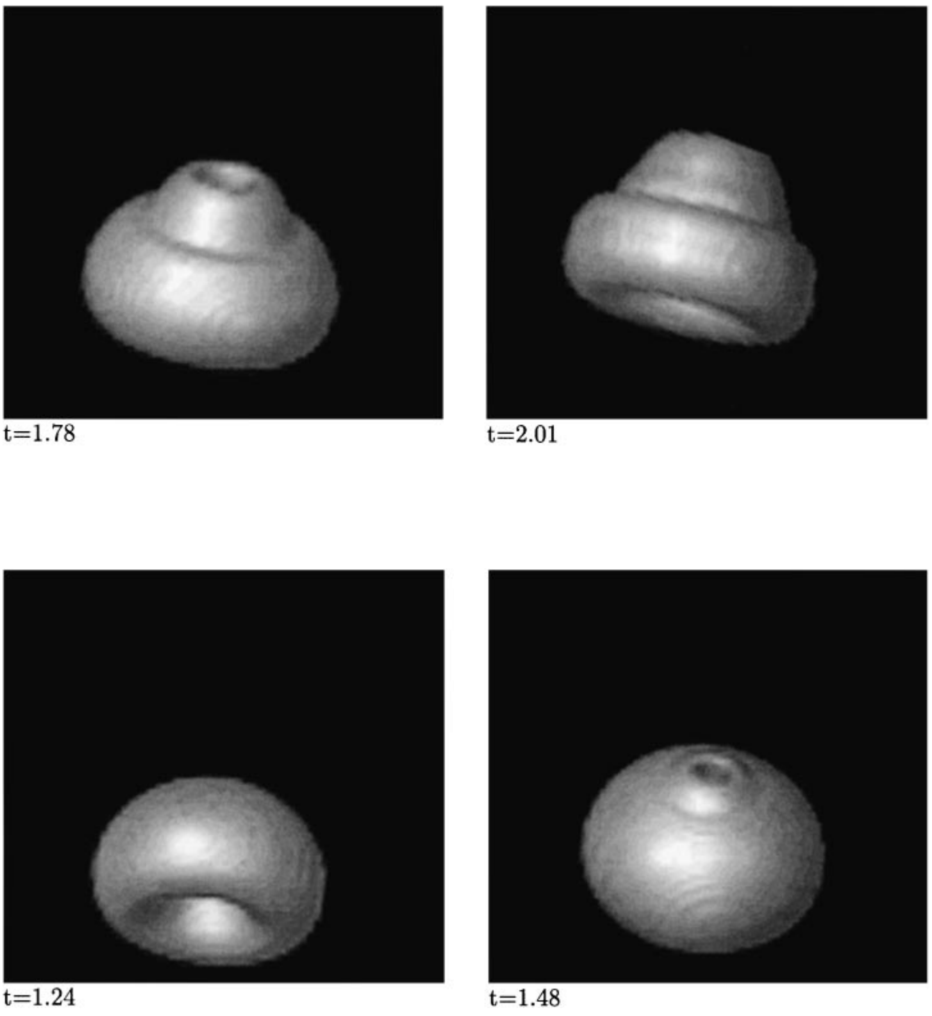

FIG. 13. Rise of inviscid air bubble in water; We = 200, effective fine grid 64 × 64 × 128.

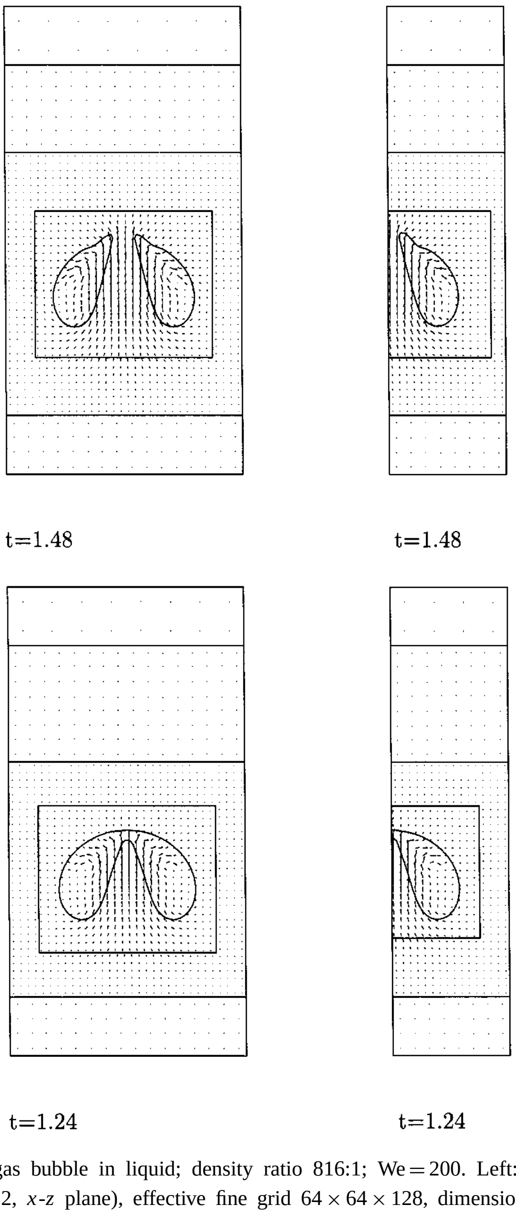

FIG. 14. Spherical gas bubble in liquid; density ratio 816:1; We = 200. Left: Cross section of three-dimensional results ($y = 2$, x-z plane), effective fine grid $64 \times 64 \times 128$, dimensions of domain: $4 \times 4 \times 8$. Right: Axisymmetric results, effective fine grid 32×128, dimensions of domain 2×8.

was used to study the impact of drops on liquid surfaces and the subsequent entrainment of an air bubble. We shall compare our results to those found in Fig. 2 of the work by Oguz and Prosperetti [36]. The initial drop radius is $L = 1.9$ mm and the drop impact velocity is $U = 1.53$ m/s. The dimensionless parameters for this problem are Re = 2550, Fr = 126, and We = 61. For our computations, the dimensionless size of the domain is 10×20, the effective fine grid resolution is 128×256, and the interfacial thickness parameter is $2\Delta x^{\ell_{\max}}$. As opposed to the computations done by [36], we do not initialize our computation with the drop already connected to the pool of water; instead we accelerate the drop towards the pool of water until it reaches the appropriate speed. The times in our computations are

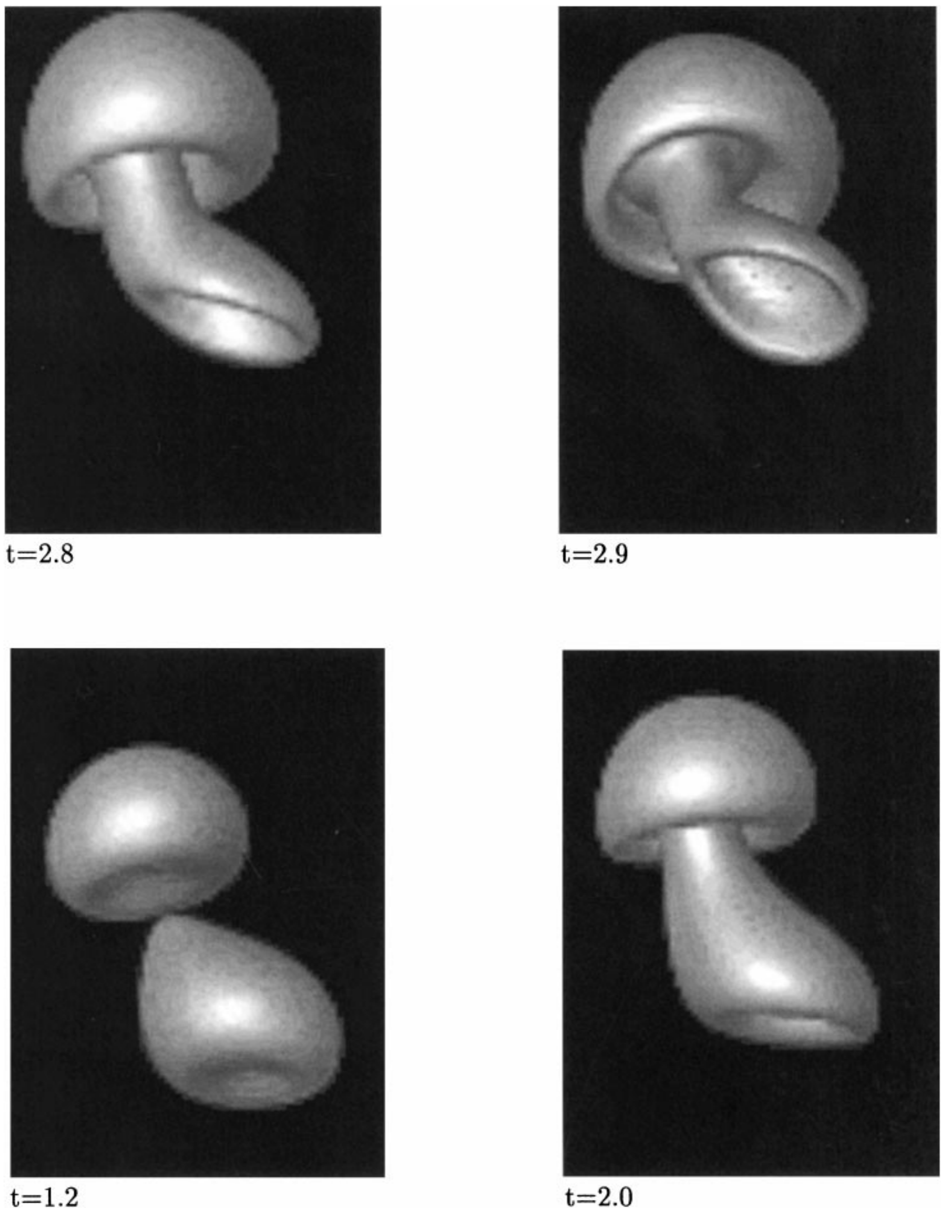

FIG. 15. Nonaxisymmetric merging of two viscous gas bubbles, effective fine grid $64 \times 64 \times 128$.

relative to the impact time of the drop hitting the pool of water. We display our results for the drop impact and subsequent entrainment in Figure 21 which is in very good agreement with the results in [36].

5.5. *Collision of Drops*

We compute the impact of two axisymmetric drops. The drops are initially driven towards each other by the force

$$f_z = \frac{1}{2}(\rho - \rho_2)\ \mathrm{sign}(z - z_c) \tag{82}$$

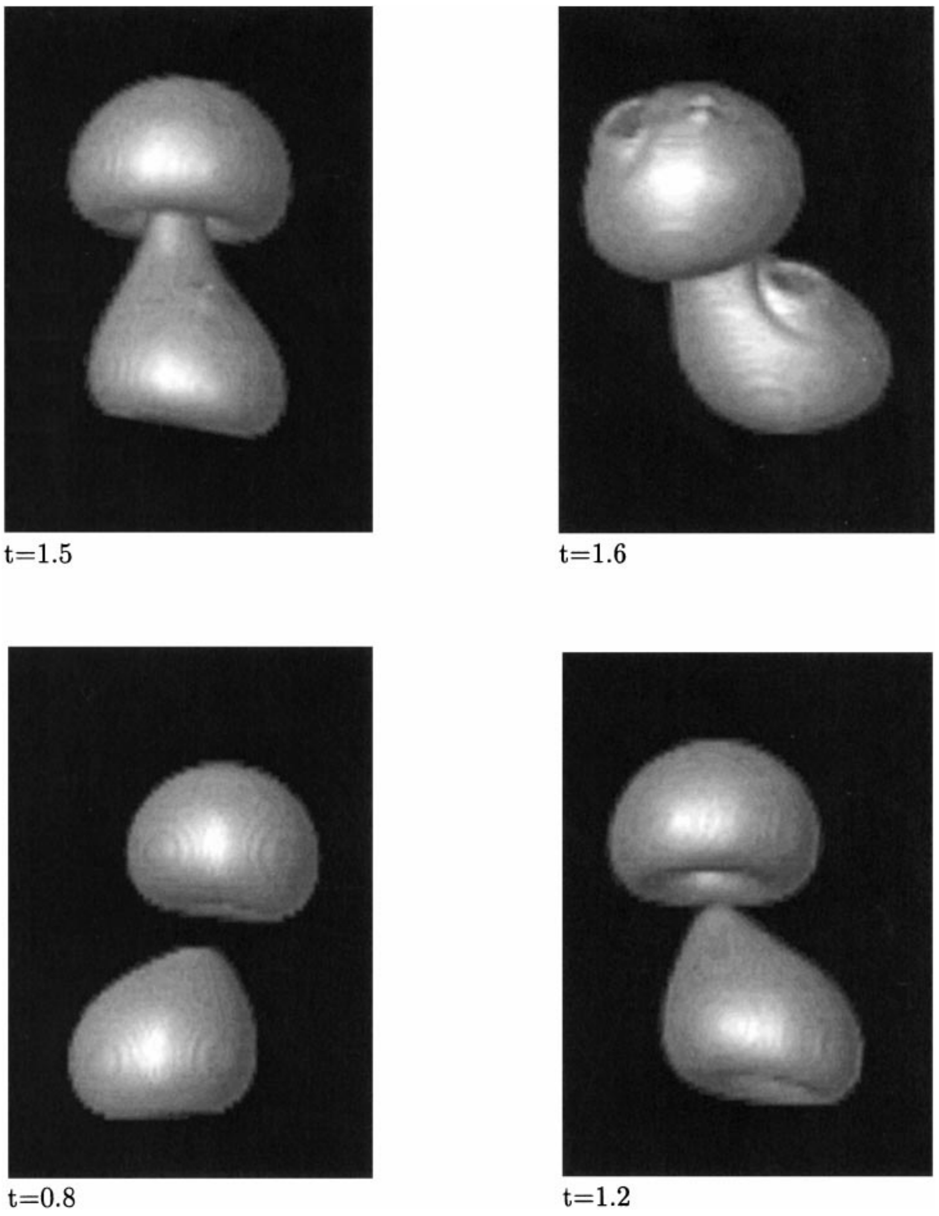

FIG. 16. Nonaxisymmetric merging of two inviscid gas bubbles, effective fine grid $64 \times 64 \times 128$.

for $t = 0 \cdots 1$; ρ_2 is the density of the gas and z_c is the point midway between the drops. The centers of the drops are initially two diameters apart. The parameters we use for this problem are $\rho_1/\rho_2 = 15$, $\mu_1/\mu_2 = 350$, $\mathrm{W} = 32$, and $\mathrm{R} = 98$. These parameters correspond to the simulation done by Nobari *et al.* [35] in Fig. 17 of their paper. For this test problem, the Weber number and Reynolds number are expressed as

$$\mathrm{W} = \frac{\rho_1 d U^2}{\sigma}, \quad R = \frac{\rho_1 U d}{\mu_1},$$

where d is the drop diameter and U is double the speed of each drop. Time is scaled by $2d/U$. In Fig. 22, we show our computation of the colliding drops in which the domain

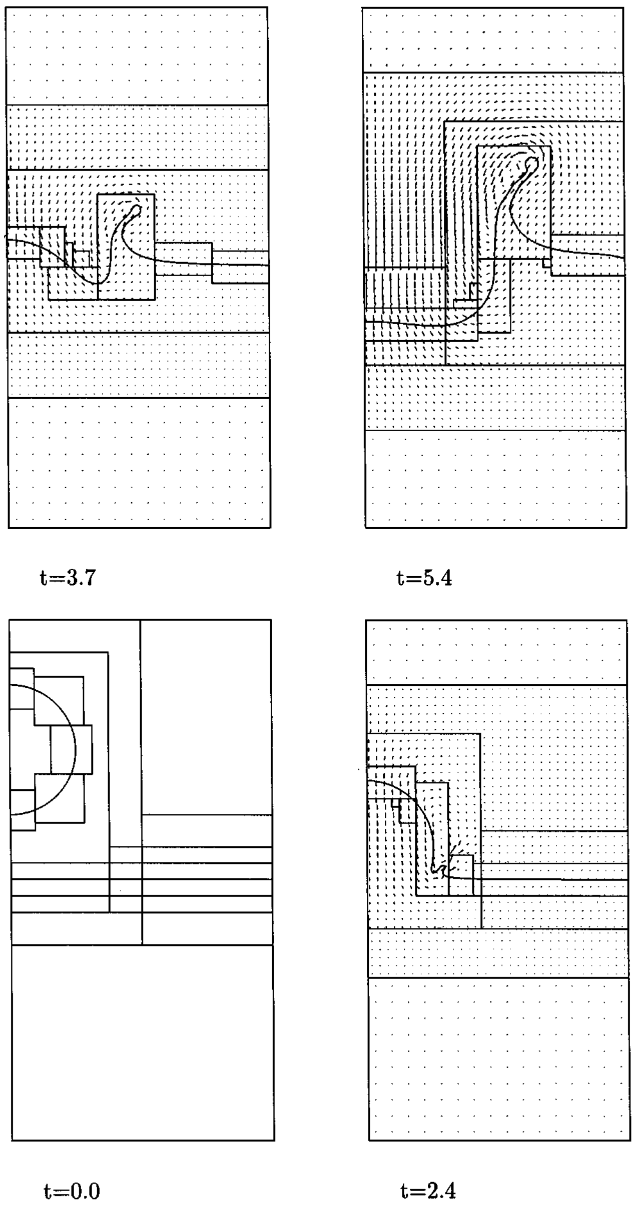

FIG. 17. Falling 1-mm spherical water drop onto pool of water; density ratio 816:1, viscosity ratio 71:1, 128×256, impact speed $U = 4.0$ m/s.

size is $3/2x6$ and the effective fine grid resolution is 64×256. As in [35], time is set to zero when the drops are one diameter apart. We remark that our results do not agree exactly with those of [35] because our drops are allowed to “rupture” immediately, whereas the computations of [35] “rupture” at $t = 0.4$.

5.6. *Fully 3D Simulations of Single Rising Gas Bubble*

In Fig. 13, we show the computation of the rise of a fully three-dimensional inviscid air bubble in water. The density ratio is 816:1 and the Weber number is 200. The dimensions

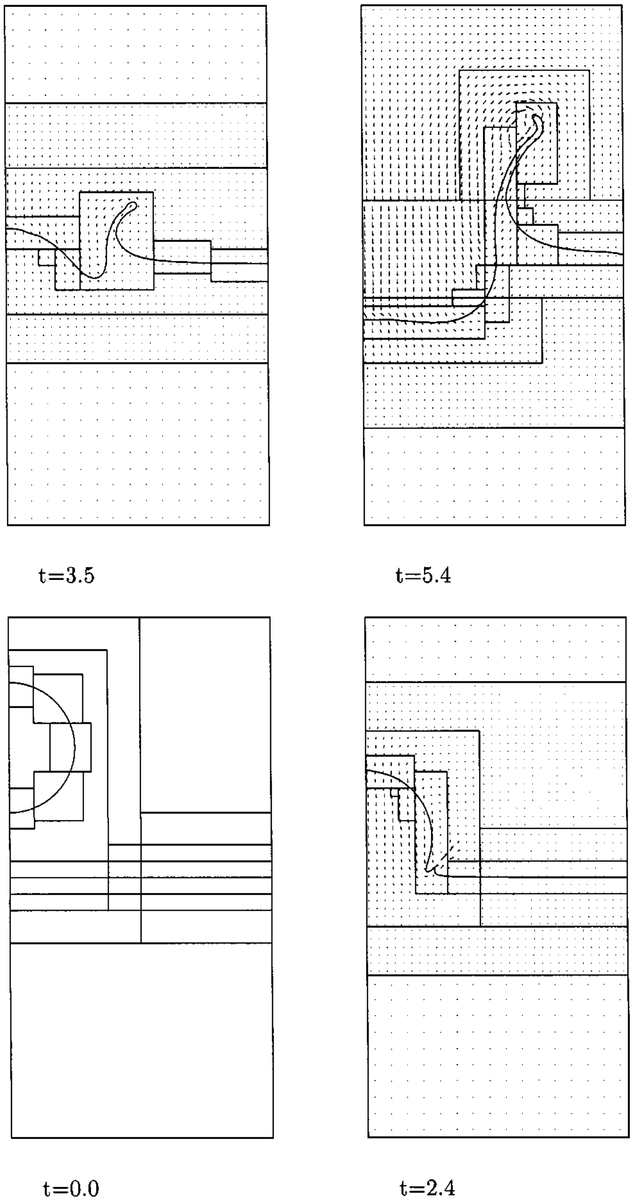

FIG. 18. Falling 1-mm spherical water drop onto pool of water; density ratio 816:1, viscosity ratio 71:1, 128×256, impact speed $U = 7.6$ m/s.

of the domain are $4 \times 4 \times 8$ and the effective number of computational cells on the finest level of adaptivity (the third level) is $64 \times 64 \times 128$. We use far-field boundary conditions on all sides of the domain. In Fig. 14, we display a cross-section of the bubble at $t = 1.24$ and $t = 1.48$ and compare these results with the results of an axisymmetric bubble problem in which the effective fine grid resolution is 32×128. We point out here that we do not have to do any extra programming in transitioning from a spherical cap bubble into a toroidal bubble.

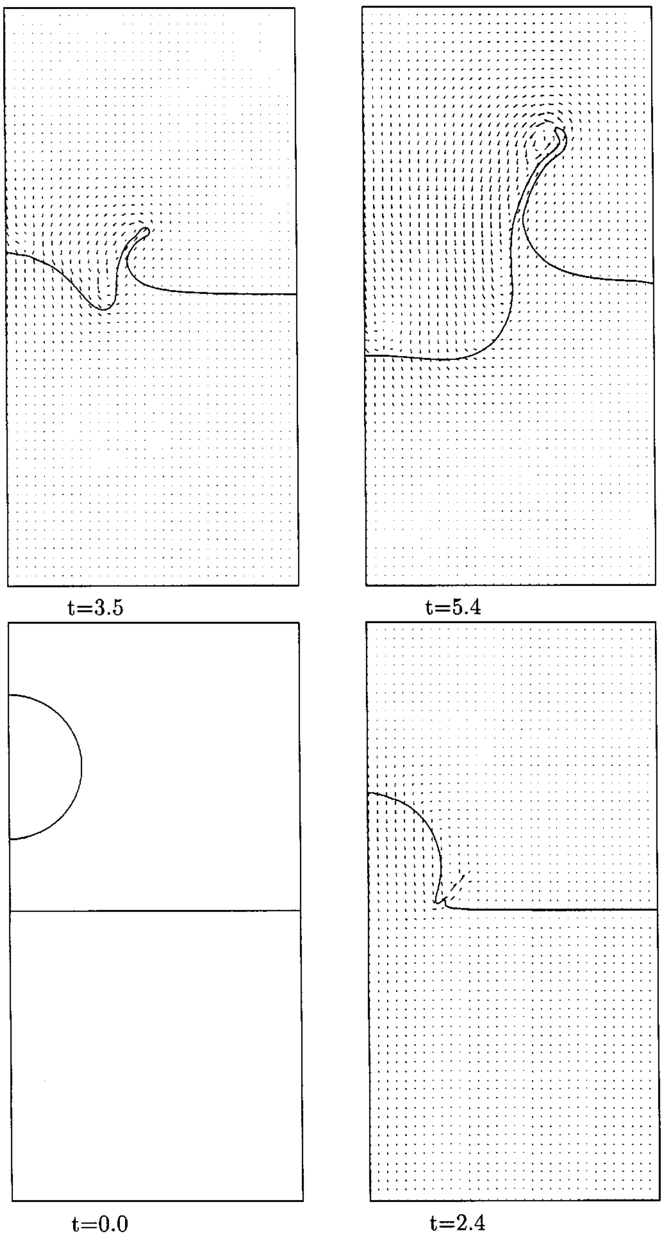

FIG. 19. Falling 1-mm spherical water drop onto pool of water; density ratio 816:1, viscosity ratio 71:1, 128×256, impact speed $U = 7.6$ m/s. Adaptive mesh refinement turned off.

5.7. *Fully 3D Simulations of the Nonaxisymmetric Merging of Two Bubbles*

For these problems, we start off with two gas bubbles whose centers are offset in the "x" direction by one bubble radii and offset in the "z" direction by 2.3 radii.

In Fig. 15, we display the interaction of two viscous gas bubbles in liquid. The density ratio is 20:1 and the viscosity ratio is 26:1. The dimensionless parameters we use for this problem are $\mathrm{W} = 50$, $\mathrm{Fr} = 1$, and $\mathrm{R} = 50^{3/4}$. These parameters correspond to Fig. 12 in [50].

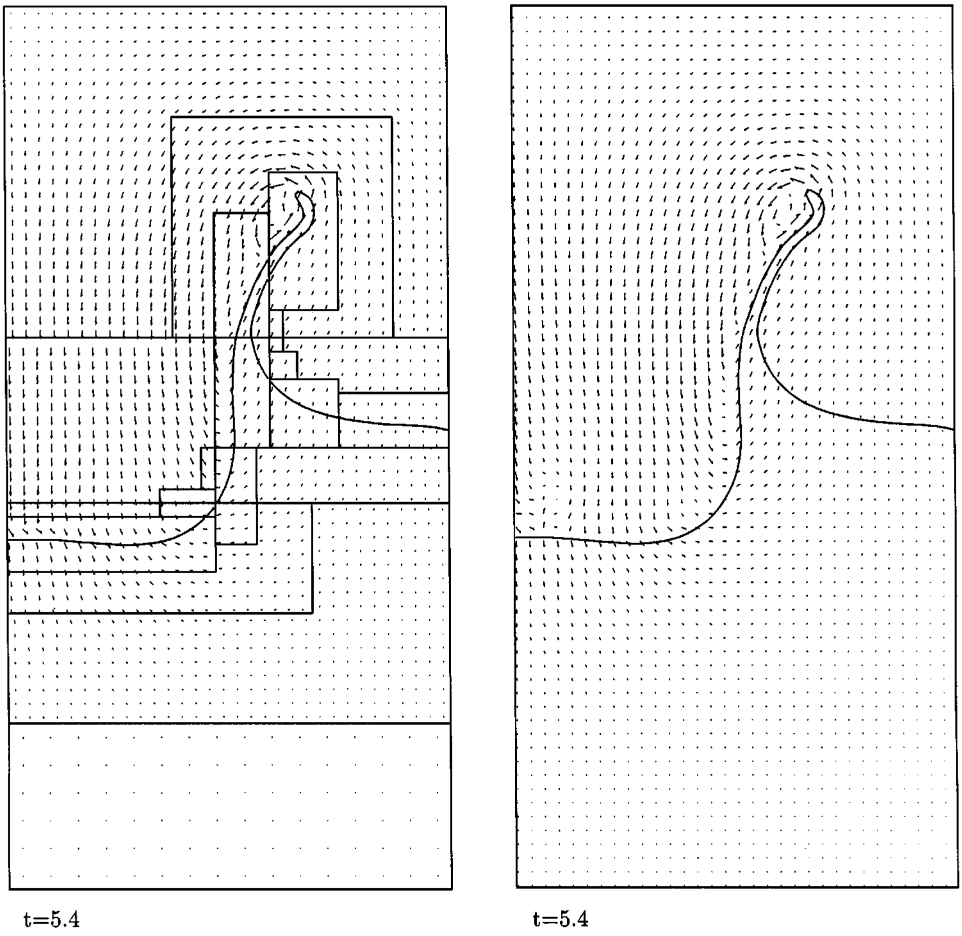

FIG. 20. Falling 1-mm spherical water drop onto pool of water; density ratio 816:1, viscosity ratio 71:1, 128×256, impact speed $U = 7.6$ m/s. Comparison between adaptive results on the left and uniform fine grid results on the right.

The dimensions of the domain are $4 \times 4 \times 8$ and the effective number of computational cells on the finest level of adaptivity (the second level) is $64 \times 64 \times 128$. We use free-slip boundary conditions on all sides of the domain. Our results agree qualitatively with those in [50]. We attribute the difference, in part, due to the fact that our bubbles were initially offset by different values than were the bubbles in [50].

In Fig. 16, we display the interaction of two inviscid gas bubbles in water. The density ratio is 816:1. As in the previous case, the dimensions of the domain are $4 \times 4 \times 8$ and the effective number of computational cells on the finest level of adaptivity (the second level) is $64 \times 64 \times 128$. We use free-slip boundary conditions on all sides of the domain.

6. CONCLUSIONS

An adaptive level set method has been presented for computing free surface flows in which large jumps in density and viscosity occur at the free surface. Surface tension forces are included in the numerical model. Adaptive mesh methodology is used to focus

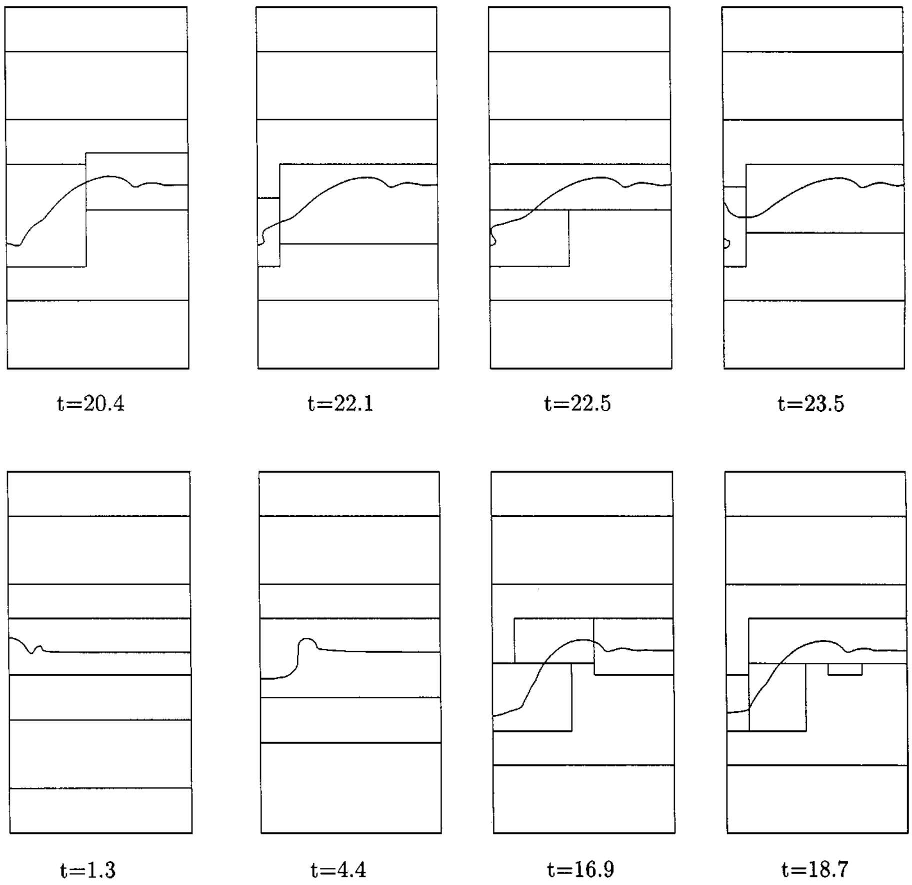

FIG. 21. Falling 1.9-mm spherical water drop onto pool of water; density ratio 816:1, viscosity ratio 71:1, 128×256, impact speed $U = 1.53$ m/s.

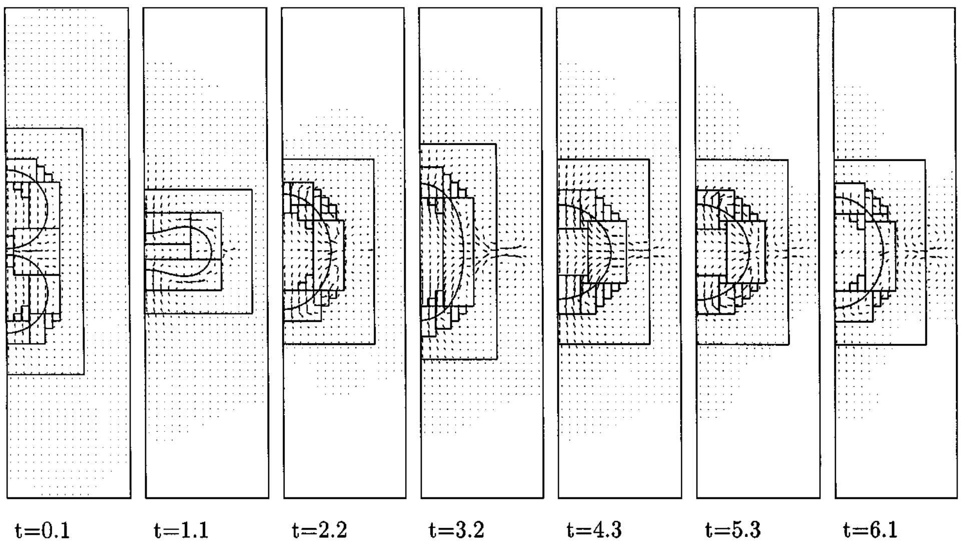

FIG. 22. Collision of axisymmetric drops; $R = 98$, $W = 32$. Time is set to zero when two drops are one diameter apart. Effective fine grid resolution is 64×256.

computational effort to regions near the free surface and in some cases focus additional attention to regions of high curvature. Examples in two and three dimensions are shown for a wide range of Reynolds numbers and Weber numbers in which the arbitrary merging and breakup of fluid mass may take place. We have validated the adaptive level set method against the bubble experiments of Hnat and Buckmaster [25], drop computations by Oguz and Prosperetti [36] and boundary integral computations in Sussman and Smereka [44]. Finally, a convergence study has been conducted in order to measure the order of accuracy for the problem of a rising inviscid air bubble in water and the problem of an oscillating droplet.

The methodology presented here is currently being extended to the problem of oil spreading under ice [46]. This problem requires the fully implicit rather than semi-implicit treatment of the viscous terms, thereby eliminating the need for a viscous time step constraint. In addition, one must impose the contact angle boundary condition at the oil/ice/water junction.

Future directions for this work include embedding the current algorithm in a framework capable of handling irregular geometries [4] and extending the algorithm to handle flows with heat transfer and vaporization [28]. These are both necessary in order to model thermal ink jet devices [2, 20], in which a vapor bubble "pushes" ink out of a jetting device. Finally, there are ongoing efforts to parallelize the adaptive incompressible flow algorithm, based on the existing block-structured data format.

REFERENCES

1. R. E. Alcouffe, A. Brandt, J. E. Dendy, and J. W. Painter, The multi-grid method for the diffusion equation with strongly discontinuous coefficients, *SIAM J. Sci. Stat. Comput.* **2**, 430 (1981).
2. R. R. Allen, J. D. Meyer, and W. R. Knight, Thermodynamics and hydrodynamics of thermal ink jets. *Hewlett-Packard J.* May (1985).
3. A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. Welcome, A conservative adaptive projection method for the incompressible Navier-Stokes equations in three dimensions, *J. Comput. Phys.*, to appear.
4. A. S. Almgren, J. B. Bell, P. Colella, and T. Marthaler, A cartesian grid projection method for the incompressible euler equations in complex geometries, *SIAM J. Sci. Comput.* **18**(5), 1289 (1997).
5. A. S. Almgren, J. B. Bell, and W. G. Szymczak, A numerical method for the incompressible Navier–Stokes equations based on an approximate projection, *SIAM J. Sci. Comput.* **17**(2), (1996).
6. J. B. Bell, M. J. Berger, J. S. Saltzman, and M. L. Welcome, Three-dimensional adaptive mesh refinement for hyperbolic conservation laws, *SIAM J. Sci. Comput.* **15**(1), 1277 (1994).
7. J. B. Bell, P. Colella, and H. M. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, *J. Comput. Phys.* **85**, 257 (1989).
8. J. B. Bell, P. Colella, and L. H. Howell, An efficient second-order projection method for viscous incompressible flow, in *10th AIAA Computational Fluid Dynamics Conference, Honolulu, June 24–27, 1991.*
9. J. B. Bell and D. L. Marcus, A second-order projection method for variable-density flows, *J. Comput. Phys.* **101**, 334 (1992).
10. M. J. Berger and P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* **82**, 64 (1989).
11. M. J. Berger and J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, *J. Comput. Phys.* **53**, 484 (1984).
12. M. J. Berger and I. Rigoustsos, *An Algorithm for Point Clustering and Grid Generation*, Technical Report NYU-501 (New York University, CIMS, 1991).
13. J. P. Best, The formation of toroidal bubbles upon the collapse of transient cavities, *J. Fluid Mech.* **251**, 79 (1993).
14. J. M. Boulton-Stone and Blake, Gas bubbles bursting at a free surface, *J. Fluid Mech.* **254**, 437 (1993).

15. J. U. Brackbill, D. B. Kothe, and C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* **100**, 335 (1992).

16. D. Chambers, D. Marcus, and M. Sussman, Relaxation spectra of surface waves, in *Proceedings of the 1995 International Mechanical Engineering Congress and Exposition, November 1995.*

17. Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher, Eulerian capturing methods based on a level set formulation for incompressible fluid interfaces, *J. Comput. Phys.* **124**, 449 (1996).

18. P. Colella, A direct Eulerian MUSCL scheme for gas dynamics, *SIAM. J. Comput.* **6**, 104 (1985).

19. P. Colella, A multidimensional second-order Godunov scheme for conservation laws, *J. Comput. Phys.* **87**, 171 (1990).

20. N. V. Deshpande, Fluid mechanics of bubble growth and collapse in a thermal ink-jet printer, in *SPSE/SPIES Electronic Imaging Devices and Systems Symposium, January 1989.*

21. A. Esmaeeli and G. Tryggvason, An inverse energy cascade in two-dimensional low reynolds number bubbly flows, *J. Fluid Mech.* **314** (1996).

22. H. Haj-Hariri, Q. Shi, and A. Borhan, Thermocapillary motion of deformable drops at finite reynolds and marangoni numbers, *Phys. Fluids* **9**, 845 (1997).

23. A. Harten, *J. Comput. Phys.* **83**, 148 (1989).

24. M. Hinatsu and H. Takeshi, Breaking waves in front of a box-shaped ship, in *2nd Japan-Korea Workshop on Ship and Marine Hydrodynamics, Osaka, Japan, 1993.*

25. J. G. Hnat and J. D. Buckmaster, Spherical cap bubbles and skirt formation, *Phys. Fluids* **19**(2), 182 (1976).

26. L. H. Howell, A multilevel adaptive projection method for unsteady incompressible flow, in *6th Copper Mountain Conference on Multigrid Methods, Copper Mountain, CO, April 4–9 1993.*

27. D. Jacqmin, *An Energy Approach to the Continuum Surface Method,* No. AIAA-96-0858, Reno, NV, 1996. [Presented at the 34th Aerospace Sciences Meeting]

28. D. Juric and G. Tryggvason, Computations of boiling flows, *Int. J. Multiphase Flow* **24**(3), 387 (1998).

29. H. Lamb, *Hydrodynamics* (Dover, New York, 1932).

30. M. S. Longuet-Higgins and E. D. Cokelet, Deformation of steep surface waves on water I: A numerical method of computation, *Proc. R. Soc. Lond. A.* **350**, 1 (1975).

31. T. S. Lundgren and N. N. Mansour, Vortex ring bubbles, *J. Fluid Mech.* **224**, 177 (1991).

32. R. B. Milne, Ph.D. thesis, An adaptive level set method, U. C. Berkeley Department of Mathematics, 1995. [LBNL Technical Report LBNL-39216]

33. M. L. Minion, A projection method for locally refined grids, *J. Comput. Phys.* **127** (1996).

34. M. L. Minion, On the stability of Godunov-projection methods for incompressible flow, *J. Comput. Phys.*, in press.

35. M. R. Nobari, Y. J. Jan, and G. Tryggvason, *Head on Collision of Drops; A Numerical Investigation*, Technical Report ICOMP-93-45, NASA ICOMP Lewis Research Center, November 1993.

36. H. N. Oguz and A. Prosperetti, Bubble entrainment by the impact of drops on liquid surfaces, *J. Fluid Mech.* **203**, 143 (1990).

37. S. Osher and J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* **79**(1), 12 (1988).

38. R. B. Pember, J. B. Bell, P. Colella, W. Y. Crutchfield, and M. L. Welcome, An adaptive Cartesian grid method for unsteady compressible flow in irregular regions, *J. Comput. Phys.* **120**, 278 (1995).

39. E. G. Puckett, A. S. Almgren, J. B. Bell, D. L. Marcus, and W. G. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, *J. Comput. Phys.* **130**, 269 (1997).

40. J. Quirk and S. Karni, On the dynamics of a shock bubble interaction, *J. Fluid Mech.* **318**, 129 (1996).

41. G. Ryskin and L. G. Leal, Numerical solution of free boundary problems in fluid mechanics. Part 1. The finite-difference technique, *J. Fluid Mech.* **148**, 1 (1984).

42. C. W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes, II, *J. Comput. Phys.* **83**, 32 (1989).

43. M. Sussman and E. Fatemi, An efficient, interface preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow, *SIAM J. Sci. Comput.*, to appear.

44. M. Sussman and P. Smereka, Axisymmetric free boundary problems, *J. Fluid Mech.* **341**, 269 (1997).

45. M. Sussman, P. Smereka, and S. J. Osher, A level set approach for computing solutions to incompressible two-phase flow, *J. Comput. Phys.* **114**, 146 (1994).

46. M. Sussman and S. Uto, *Computing Oil Spreading Underneath a Sheet of Ice,* Technical Report CAM Report 98-32, University of California, Los Angeles, July 1998.

47. W. G. Szymczak, J. Rogers, J. M. Solomon, and A. E. Berger, A numerical algorithm for hydrodynamic free boundary problems, *J. Comput. Phys.* **106**, 319 (1993).

48. O. Tatebe, The multigrid preconditioned conjugate gradient method, in *6th Copper Mountain Conference on Multigrid Methods, Copper Mountain, CO, April 4–9 1993.*

49. H. S. Udaykumar, M. M. Rao, and W. Shyy, Elafint—A mixed Eulerian–Lagrangian method for fluid flows with complex and moving boundaries, *Int. J. Numer. Methods Fluids.* **22**(8), 691 (1996).

50. S. O. Unverdi and G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, *J. Comput. Phys.* **100**, 25 (1992).

51. P. Wesseling, *An Introduction to Multigrid Methods* (Wiley, New York, 1992).